UNIVERSIDAD DE LA RIOJA

TRABAJO FIN DE ESTUDIOS

Título

Propuesta de sistema inteligente de detección de accidentes para el automóvil con avisos de emergencia

Autor/es

EDER VILLAR MARTÍNEZ

Director/es

JAVIER ESTEBAN VICUÑA MARTÍNEZ

Facultad

Escuela Técnica Superior de Ingeniería Industrial

Titulación

Grado en Ingeniería Electrónica Industrial y Automática

Departamento

INGENIERÍA ELÉCTRICA

Curso académico

2018-19

publicaciones.unirioja.es
E-mail: publicaciones@unirioja.es

UNIVERSIDAD DE LA RIOJA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL

TRABAJO DE FIN DE GRADO

TITULACIÓN: Grado en
Ingeniería Electrónica Industrial y Automática

CURSO: **2018/2019** **CONVOCATORIA: SEPTIEMBRE**

TÍTULO:

Propuesta de sistema inteligente de detección de accidentes para el automóvil con avisos de emergencia

ESTUDIANTE: Éder Villar Martínez

TUTORES/AS: Javier Esteban Vicuña Martínez

DEPARTAMENTO: Ingeniería Eléctrica

Resumen

En este proyecto se realiza un estudio sobre un posible sistema de emergencia que ayudará a los ocupantes de un coche cuando se haya producido un accidente. El principal objetivo de este trabajo es el estudio acerca de un sistema que, una vez instalado en cualquier coche, sea capaz de aumentar la probabilidad de supervivencia de los ocupantes en caso de accidente. Además, el sistema tendría la capacidad de preservar intacto tras el suceso para así ser capaz de extraer la información que contenga y realizar un estudio de lo sucedido, consiguiendo así una mejora en los vehículos en el ámbito de la seguridad.

Este TFG se inicia con el análisis de otros sistemas que ya existen en el mercado y que persiguen el mismo objetivo que tiene este estudio. Posteriormente, se exponen las especificaciones que debería tener el sistema que vamos a estudiar y, además, se hace alguna pequeña toma de decisiones entre alguna posible solución a los problemas que se presentan. En esta toma de decisiones se ha incluido alguna ventaja y desventaja de cada solución planteada y se ha elegido la que se cree más conveniente. Además, se han planteado distintas sugerencias de software a utilizar para llevar a la realidad este estudio.

Posteriormente se ha ido planteando el sistema descrito, pero a un nivel más detallado. En este bloque del documento se explican apuntes teóricos, necesarios para la comprensión de algún punto, y se exponen soluciones hardware para llegar a construir un prototipo.

Este sistema cuenta con la capacidad de detectar un accidente y clasificar el suceso. La gravedad del acontecimiento se divide en cuatro niveles de gravedad yendo desde funcionamiento normal hasta accidente grave. Ya realizada la clasificación y teniendo en cuenta la gravedad del accidente, este dispositivo será capaz, tanto de avisar a los coches cercanos del suceso ocurrido para evitar otro posible accidente, como de contactar con los servicios de emergencia enviándoles un pequeño informe de todo lo ocurrido. Por otro lado, toda la parte hardware que incluye este sistema está protegida para evitar su destrucción.

Por último, se exponen distintas vías de mejora para el sistema planteado. Se explora la posibilidad de diseñar un chip que incluya todo el hardware que necesitamos, e incluso, factibles soluciones a posibles problemas que podrían ocurrir en el caso de que algún componente fallase. Por ejemplo, un sensor o la toma de decisiones que realiza nuestro microprocesador.

Abstract

This project centres itself around a study on a potential emergency system, based around helping the occupants of a car in case of an accident. The primary objective of this study concerns the study of a system that, when installed in any car, increases the probability of survival of the occupant in case of an accident. Additionally, this system would have the capacity to preserve its integrity during the incident, enabling the extraction and study of the data it contains, with the overall objective of making improvements on the field of safety.

This thesis opens on an analysis of other, previously existing, systems, which are commercially available and pursue the same objective. It continues submitting the specifications required for the system, as well as presenting certain decision-making processes followed for the given problems. The decision-making processes include advantages and disadvantages of each solution. Additionally, some suggestions have been offered as to the recommended software necessary to put this study to practice.

Subsequently, the system has been laid out with further detail, accompanied with notes that explain certain knowledge bases necessary for the comprehension of certain points, explaining, moreover, the hardware specifications required for the development of a prototype.

This system achieves the ability to detect an accident and classify the incident. The degree of gravity of said incident is classified into four levels, ranging from regular functionality to severe accident. Once the incident is classified, and, considering its severity, the device will be capable of warning other nearby vehicles, as well as notifying the emergency services, forwarding a summary report on the incident. Considering its purpose, the entirety of the hardware of this system is protected to prevent its destruction.

Finally, various improvements are explored for the system, including the possibility of designing a chip that encompasses the totality of the required hardware, and even, possible solution to problems arising from the failure of certain components. For example, a sensor or the decision-making carried out by the microprocessor.

1 Índice

1.1 Índice del documento

1.2 Índice de figuras

2 Memoria

2.1 Presentación

Este trabajo ha sido realizado por D. Éder Villar Martínez, alumno del cuarto curso del Grado en Ingeniería Electrónica Industrial y Automática. Se presenta al objeto de la obtención del título de Graduado en Ingeniería Industrial por Universidad de La Rioja.

2.2 Objeto

El principal objetivo de este proyecto es el estudio de una propuesta para un sistema de seguridad a instalar en vehículos que permita en caso de accidente, identificar y clasificar el tipo de accidente sufrido y actuar en consecuencia desplegando un conjunto de alertas de seguridad y de petición de ayuda, si fuese necesario.

Esta toma de decisión será llevada a cabo por un sistema basado en inteligencia artificial que habrá sido previamente entrenado con un conjunto suficiente de datos de accidentes, para en caso de accidente, inferir una clasificación entre cuatro grados de accidentes, en función de su gravedad. Los datos con los que el modelo clasifica son obtenidos, principalmente, de los datos provenientes del bus CAN del vehículo.

Como consecuencia a la clasificación obtenida del grado de accidente sufrido, el sistema ejecutará automáticamente un conjunto de acciones proporcionales. Estas acciones pueden ir desde la inacción, hasta el aseguramiento del vehículo accidentado, la alerta a otros vehículos próximos, y/o el aviso a los servicios de emergencia.

2.3 Objetivos parciales o etapas

Para cumplir el objetivo principal establecido para el TFG, ha sido necesario completar un conjunto de etapas, tareas u objetivos parciales, que se enumeran a continuación:

- Análisis de las especificaciones del sistema a diseñar.
- Estudio de funcionamiento de las comunicaciones entre sensores. Esto es el bus CAN.
- Estudio y comprensión del Deep Learning y las redes neuronales. Tipos, topología, entrenamiento, fundamentos matemáticos, etc.
- Estudio de la comunicación por radiofrecuencia y elección de módulos de emisión y recepción para comunicación entre vehículos.

- Búsqueda de un sistema basado en microprocesador que sea capaz de realizar todas estas operaciones.
- Búsqueda de un dispositivo que permita la comunicación con los servicios de emergencia, 112 en Europa.
- Análisis de posibles mejoras.

2.4 Alcance

Debido a que este TFG se apoya en datos privados de empresas los cuales no nos son accesibles, como datos de colisiones o de tramas CAN que utiliza cada coche, este proyecto no contempla:

- El desarrollo de un sistema con las características y funcionalidades mencionadas que sea perfectamente funcional para un vehículo en concreto o para todo el conjunto de vehículos.
- El desarrollo de un prototipo que funcione parcialmente con los objetivos planteados debido a las mismas limitaciones que se mencionan con anterioridad.
- El desarrollo de alguna parte que se expone en este documento ya que es un documento meramente de estudio.

Por otra parte, este documento si considera:

- La realización de todos los puntos que se han mencionado en el apartado anterior, 2.3.

2.5 Justificación

La seguridad en vehículos actuales es una característica prioritaria a la que la industria del automóvil dedica multitud de recursos. El objetivo es el diseño y fabricación de modelos de automóvil cada vez más seguros y fiables. A los vehículos se les dota de sistemas específicos que de forma activa y/o pasiva ayudan a evitar que los accidentes se produzcan (reducir su probabilidad) o de reducir los daños físicos a los ocupantes, en caso de que se produzca un accidente.

En sus inicios, se dotó a los vehículos sistemas como el cinturón de seguridad: un arnés diseñado para mantener en su asiento al ocupante en caso de colisión y además hasta el día de hoy, es el único elemento de seguridad pasivo cien por cien fiable. Este comenzó a utilizarse en aeronaves por el año 1930 y posteriormente se implementó en los automóviles. (Wikipedia 2019)

Más adelante apareció el llamado Airbag que consiste en una bolsa de aire que se hincha a alta velocidad como consecuencia de una colisión. Así, cuando los ocupantes del vehículo son propulsados hacia la dirección contraria

al sentido del impacto, esta bolsa de aire amortigua la mayor parte del impacto. Al principio, sólo se instalaba airbag del conductor, justo en el volante, pero conforme fueron avanzando los años se implementaron airbags en más zonas como en el salpicadero en la zona del copiloto, en la zona de las puertas para las colisiones laterales, en los pies del conductor, etc.

Otra tecnología que se desarrolló y que sigue desarrollándose para mejorar la seguridad de los ocupantes en caso de accidente es la estructura de deformación programada o autoportante. Después de realizar muchas pruebas con carrocerías rígidas, se dieron cuenta que estas no eran las mejores ya que en caso de accidente, toda la fuerza del impacto recaería sobre los ocupantes por lo que, las posibilidades de supervivencia son bajas. Debido a esto, se crearon las carrocerías deformables. Estas carrocerías son capaces de transformar la energía cinética del impacto en energía de deformación, es decir, esta carrocería estaría absorbiendo parte del impacto por lo que se produce una disminución en la fuerza de desaceleración que sufren los ocupantes. (Todoautos 2012)

En los últimos años se han desarrollado mucho los coches inteligentes, que son capaces de anticiparse a un accidente e intentar evitarlo. Son capaces, por ejemplo, de esquivar un coche cuando nos dirigimos hacia él sin control, frenar automáticamente en caso de que vayamos a atropellar a un peatón o una colisión por alcance con otro vehículo, incluso el propio coche te avisa de parar a descansar en el caso de que vea signos de fatiga en el conductor. Sistemas ABS, sistemas de control antiderrapaje, sistemas que evitan el cambio involuntario de carril, lectura automática de señales, etc.

Pero aún con todos estos sistemas de seguridad y todos los años que llevan de desarrollo a sus espaldas, los accidentes siguen produciéndose, a pesar de que hayan realizado su cometido correctamente.

Por citar un ejemplo, el conductor puede no estar en condiciones físicas de abandonar el vehículo por su propio pie, por lo que necesita ayuda externa y en muchos casos, por desgracia, esa ayuda externa llega cuando ya es demasiado tarde. O, por ejemplo, que, por la visibilidad y posición del vehículo accidentado, éste provoque la colisión de otros vehículos que circulan por la misma vía.

La idea de este TFG surge desde ese punto; señalizar y alertar a otros conductores de la existencia de un vehículo accidentado para evitar accidentes en cadena y prestar auxilio necesario a los ocupantes del vehículo accidentado en el menor tiempo posible, aumentando así las posibilidades de supervivencia de los accidentados.

2.6 Esquema general del documento

En la primera parte del documento se han descrito los objetivos de este, así como el alcance y la justificación de este. En la siguiente parte se ha descrito en qué punto se encuentran tecnologías como esta, así como otras tecnologías de seguridad que ya se usan en los automóviles.

Después, se ha presentado el sistema que se va a estudiar. Primero se han expuesto las especificaciones y luego se ha hecho un resumen general de las características que deberá poseer.

Luego se ha hecho una explicación más detallada de los componentes que posee este sistema y de recomendaciones para llevarlo a la realidad. Además, también se ha hecho algún apunte teórico en esta parte.

Por último, se han expuesto las condiciones necesarias para llevar a cabo este proyecto y luego se ha hecho una valoración económica de todo lo estudiado.

2.7 Antecedentes: "Estado del Arte"

En los vehículos comercializados actualmente se dispone de diferentes sistemas que se fundamentan en técnicas diversas que permiten evitar accidentes, por ejemplo, el sistema de frenado automático en caso de atropello, aviso al conductor cuando pisa las líneas de la carretera, aviso al conductor cuando de cambio involuntario de carril, aviso al conductor en caso de que padezca síntomas de insomnio o incluso los coches más avanzados en este aspecto son capaces de anticiparse al suceso del accidente e intentar evitarlo en la mayor medida posible.

Las mejoras en seguridad suelen desarrollarse por los fabricantes e inicialmente instalarse en los vehículos de más alta gama. Con posterioridad, muchos de estos sistemas se convierten en características de base que se incorporan a los vehículos de menores prestaciones o precio y se convierten en sistemas obligatorios en todos los automóviles comercializados, como, por ejemplo, los cinturones de seguridad, los airbags, sistema abs, etc.

Todos los sistemas de seguridad que persiguen evitar en lo posible los accidentes dejar de ser útiles en el momento en que, a pesar de todo, el accidente se ha producido. Tras el accidente de un vehículo, en función de la forma en que ha tenido lugar y ha quedado en la calzada, la forma segura de proceder comienza con señalizar el accidente para evitar choques en cadena, colocación de triángulos, "flash lights", chalecos reflectantes para abandonar el vehículo, etc. solicitud de ayuda, atención a los heridos en una zona segura, etc.

A este respecto y en relación con la situación post-accidente, a partir del 1 de abril de 2018, se ha hecho obligatorio instalar en todos los vehículos fabricados

desde ese día, un sistema para vehículos post-accidentados que simplifique y reduzca en lo posible el tiempo que transcurre desde que ha tenido lugar el accidente, hasta que las víctimas son atendidas por los servicios de emergencia. A este sistema se le conoce como sistema eCall.

1.1.1 Sistema eCall

Este es un sistema de llamada de emergencia que está integrado en el vehículo. Este sistema es capaz de llamar al servicio de emergencias europeos de forma manual, si pulsamos un botón de SOS que incorpora el vehículo, o automática en caso de accidente. Esta llamada funciona a través de la red inalámbrica móvil, es decir, es necesaria una tarjeta sim.

En caso de accidente, el sistema eCall se activa y hace una llamada a los servicios de emergencia que en Europa son el número 112. Esta llamada lleva lo que se denomina “eCall flag” que permite discriminar entre otras llamadas de emergencia para derivar esta al departamento que corresponda.

Cuando se establece conexión se envían datos como la posición GPS, hora del accidente, número de pasajeros, dirección y sentido del vehículo e incluso información técnica del propio vehículo. Además, los servicios de emergencia pueden ponerse en contacto con el vehículo para asegurarse si es necesario desplegar los servicios de emergencia.

Este sistema también puede ser activado de forma manual mediante un botón físico de “SOS” que se instala dentro del vehículo para que los ocupantes puedan accionarlo. El sistema realizaría las mismas acciones que si se hubiera activado de forma automática. (Wikipedia 2019)

Figura 1: Ejemplo de botón SOS instalado en un automóvil para desencadenar la llamada eCall

Este sistema es de gran ayuda para socorrer a las víctimas en caso necesario ya que los primeros minutos después del accidente son claves para poder rescatar a las víctimas con vida.

Las estimaciones realizadas dicen que gracias a este sistema se conseguirán reducir los tiempos de reacción de los sistemas de emergencia en hasta un 50%. Esto se traduce en unas 1500 muertes menos al año.

Antes de hacerse obligatorio este sistema, Opel presentó su sistema llamado Open OnStar, que aparte de ser un sistema que incorpora WiFi para los ocupantes del vehículo y que te proporciona avisos sobre mantenimiento y otras características del vehículo entre otras funcionalidades, incorpora un sistema de emergencia para accidentes diseñado y gestionado por la propia marca. (Nogales 2018)

1.1.2 Sistema de emergencia de Opel OnStar

Este sistema de Opel es muy similar al obligatorio eCall que se instala en los vehículos, con la salvedad de que, en este caso, cuando se activa este servicio de la propia compañía Opel la que recibe la llamada y gestiona la solicitud de emergencia. Cuando esta llamada es aceptada, los servicios de emergencia se pondrán de camino al vehículo accidentado y los operadores de Opel se encargan de recopilar todos los datos posibles para facilitarles a los servicios de emergencia. Esta central de Opel está disponible 24 horas al día los 365 días del año.

Al igual que el sistema eCall, este servicio será accionado de manera automática en caso de que el coche detecte colisión, o que los ocupantes de este accionen manualmente el botón de “SOS” (Redacción 2016)

2.8 Normas y referencias aplicables

El CAN del coche, tanto sea CAN o CAN FD (flexible data-rate), debe regirse por las normas:

- ISO/DIS 11898-5 2007
- ISO/DIS 11898-6 2013
- ISO 11898-1:2015, Part 1: Data link layer and physical signalling
- ISO 11898-2:2016, Part 2: High-speed medium access unit
- ISO 11898-3:2006. Part 3: Low-speed, fault-tolerant, medium-dependent interface.
- ISO 11898-4:2004, Part 4: Time-triggered communication.
- ISO 11898-5:2007, Part 5: High-speed medium access unit with low power mode

- ISO 11898-6:2013, Part 6: High-speed medium access unit with selective wake-up functionality
- ISO 16845:2016, Conformance test plan

Emisión de radiofrecuencias:

ISO/IEC 18000-7:2009 defines the air interface for radio frequency identification (RFID) devices operating as an active RF tag in the 433 MHz band used in item management applications. It provides a common technical specification for RFID devices that can be used by ISO technical committees developing RFID application standards.

- 2009 is intended to allow for compatibility and to encourage inter-operability of products for the growing RFID market in the international marketplace.
- 2009 defines the forward and return link parameters for technical attributes including, but not limited to, operating frequency, operating channel accuracy, occupied channel bandwidth, maximum power, spurious emissions, modulation, duty cycle, data coding, bit rate, bit rate accuracy, bit transmission order, and, where appropriate, operating channels, frequency hop rate, hop sequence, spreading sequence, and chip rate.
- 2009 further defines the communications protocol used in the air interface.

2.9 Disposiciones legales y normativa aplicada

Ley Orgánica 10/1991, de 8 de abril, de publicidad electoral en emisoras municipales de radiodifusión sonora.

Ley 7/2010, de 31 de marzo, General de la Comunicación Audiovisual.

En caso de encontrarse en la comunidad autónoma de La Rioja, también se tiene que tener en cuenta el siguiente decreto:

- **Decreto 64/2012, de 9 de noviembre,** por el que se regulan los servicios de comunicación audiovisual y el Registro de Prestadores en el ámbito de la Comunidad Autónoma de La Rioja.

Reglamento de telecomunicaciones:

- **Ley 9/2014, de 9 de mayo,** General de Telecomunicaciones, constituye la norma básica que desarrolla a nivel nacional los objetivos de la Agenda Digital estableciendo un marco legal armonizado que facilite el desarrollo

de las infraestructuras de las telecomunicaciones y la puesta a disposición de los ciudadanos de servicios de calidad a precios competitivos.

- **Ley 9/2014, de 9 de mayo** General de Telecomunicaciones, dedica su título V a la regulación del espectro radioeléctrico, declarándose bien de dominio público, cuya titularidad y administración corresponden al Estado.
- **Ley 9/2014, de 9 de mayo,** General de Telecomunicaciones que, además de reducir los trámites administrativos, tiene en cuenta la singularidad de estos servicios y la confidencialidad o la urgencia, que, en determinados casos, puede estar asociada a los mismos y que hace que tengan un tratamiento especial.
- **Apartado b del artículo 61 de la Ley 9/2014, de 9 de mayo,** General de Telecomunicaciones, se incorpora a este reglamento el procedimiento de control e inspección de los niveles únicos de emisión radioeléctrica tolerable y que no supongan un peligro para la salud pública.

2.10 Programas y software de desarrollo recomendado

Para llevar a cabo este proyecto se necesitan dos softwares. El primero es el necesario para programar el microprocesador que se encargará de la adquisición de datos, la toma de decisiones y comandar todos los módulos conectados a él. El segundo será el que utilicemos para crear y entrenar la red neuronal artificial que tomará las decisiones dentro del microprocesador.

El microprocesador que utilizaremos va montado en una placa llamada ChipKIT. Esta placa es compatible con Arduino y es la propia marca la que nos ofrece el Arduino IDE con el que la podemos programar. Se trata de un software gratuito y con todas las funciones y librerías necesarias para poder programar placas Arduino. No obstante, existen otros softwares que también son capaces de programar estas placas, como por ejemplo Processing, Simulink, etc.

Se recomienda la utilización de este software, Arduino IDE ya que está totalmente preparado para programar este tipo de placas y es muy sencillo de utilizar. Además, es totalmente gratuito, incluso las librerías que se utilizarán.

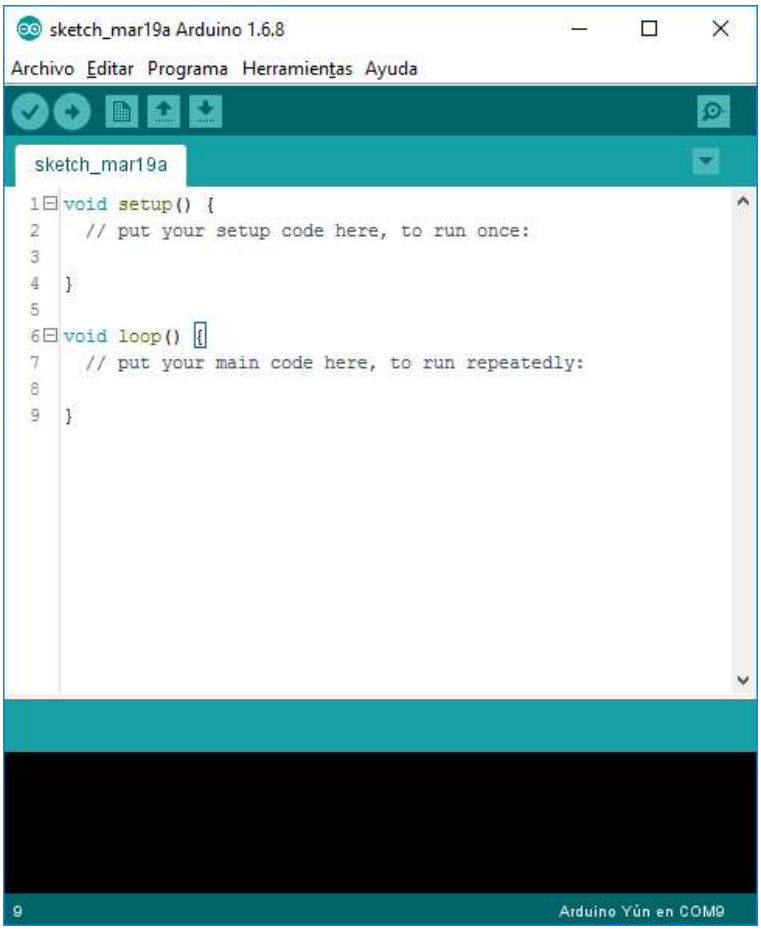


Figura 2: Interfaz de usuario de Arduino IDE

El segundo software debe ser capaz de diseñar, crear y entrenar redes neuronales artificiales. Para esto existen muchos softwares especializados o incluso, se puede crear y entrenar una red neuronal desde cero, sin utilizar programas ni librerías, únicamente programando texto estructurado. Esta última opción es poco viable ya que es muy complejo crearlo desde cero y, además, existen muchas herramientas que nos lo ponen más sencillo.

Una librería muy famosa para texto estructurado es TensorFlow. Se trata de una librería de código abierto para crear proyectos de inteligencia artificial. Se trata de una herramienta muy potente creada por Google.

Otro software y el que más recomendamos es Deep Learning Toolbox for MatLab. Se trata de una herramienta para el software de cálculo MatLab que nos permite diseñar, crear y entrenar distintos tipos de redes neuronales de una forma parcialmente gráfica. Además, tiene modelos que ya están pre entrenados y que se pueden utilizar.

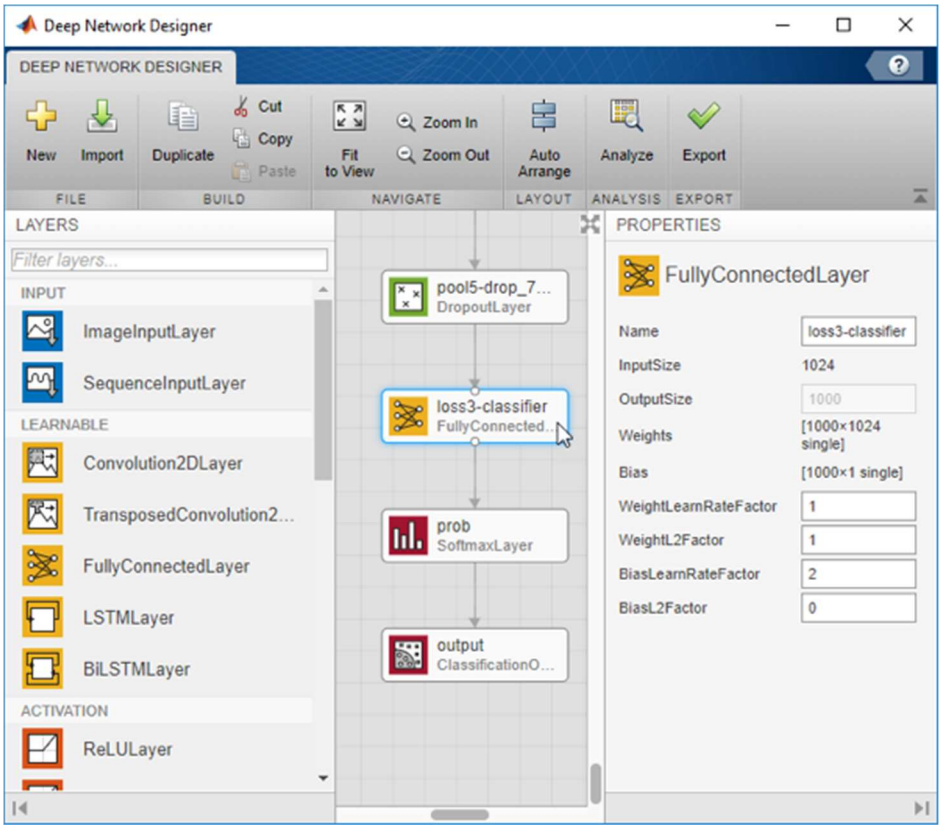


Figura 3: Interfaz de usuario de MatLab con la toolbox de Deep Learning

2.11 Bibliografía y enlaces web

Bejerano, Pablo G. 2017. https://blogthinkbig.com/diferencias-entre-machine-learning-y-deep-learning (último acceso: Julio de 2019).

Calvo, Diego. 2018. http://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/ (último acceso: Julio de 2019).

—. 2018. http://www.diegocalvo.es/red-neuronal-recurrente/ (último acceso: Julio de 2019).

Caparrini, Fernando Sancho. 2018. http://www.cs.us.es/~fsancho/?e=77 (último acceso: Julio de 2019).

CSV, Dot. 2018. https://www.youtube.com/watch?v=A6FiCDoz8_4&t=235s (último acceso: Agosto de 2019).

—. 3 de Octubre de 2018. https://www.youtube.com/watch?v=eNIqz_noix8&t=513s (último acceso: Agosto de 2019).

—. 14 de Octubre de 2018. https://www.youtube.com/watch?v=M5QHwkkHgAA&t=61s (último acceso: Agosto de 2019).

Digilent. 2015. https://reference.digilentinc.com/_media/chipkit_max32:chipkit_max32_rm.pdf (último acceso: Agosto de 2019).

Justiniano, Daniela Alvarado. 2018. https://biblioteca.unirioja.es/tfe_e/TFE004359.pdf (último acceso: Agosto de 2019).

Kent, Gabriel. 2019. https://blog.adext.com/machine-learning-guia-completa/ (último acceso: Julio de 2019).

KZgunea. 2017. http://kzgunea.blog.euskadi.eus/blog/2017/03/31/geolocalizacion-que-es/ (último acceso: Agosto de 2019).

Llamas, Luis. 2019. https://www.luisllamas.es/comunicacion-inalambrica-en-arduino-con-modulos-rf-433mhz/ (último acceso: Agosto de 2019).

marketINhouse. 2019. https://www.marketinhouse.es/que-es-la-geolocalizacion/ (último acceso: Agosto de 2019).

Nogales, MArio. 2018. https://noticias.coches.com/consejos/ecall-que-es-sistema-obligatorio-europa/287027 (último acceso: Julio de 2019).

Olmo, Lara. 2016. https://www.ticbeat.com/tecnologias/que-diferencia-hay-entre-deep-learning-inteligencia-artificial-y-machine-learning/ (último acceso: Julio de 2019).

Osés, Alejandro García. 2015. https://academica-e.unavarra.es/bitstream/handle/2454/19115/TFG%20Dise%C3%B1o%20de%20una%20Red%20Can%20bus%20-%20Alejandro%20Garc%C3%ADa%20Os%C3%A9s.pdf?sequence=1&isAllowed=y (último acceso: Junio de 2019).

Pérez Porto, Julián, y Ana Gardey. 2010. https://definicion.de/sensor/ (último acceso: Agosto de 2019).

Redacción. 2016. https://noticias.coches.com/noticias-motor/opel-onstar/196906 (último acceso: Julio de 2019).

Rodrígez, Txema. 2018. https://www.xataka.com/robotica-e-ia/machine-learning-y-deep-learning-como-entender-las-claves-del-presente-y-futuro-de-la-inteligencia-artificial (último acceso: Julio de 2019).

Sanseviero, Omar. 2018. https://medium.com/ai-learners/ai-en-3-minutos-tipos-de-machine-learning-945b708ac78 (último acceso: Julio de 2019).

Todoautos. 2012. http://www.todoautos.com.pe/portal/mecanica/482-automotriz/2399-carroceria-deformable-mecanica-auto (último acceso: Agosto de 2019).

Torra, Vicenç. 2011. http://www.fgcsic.es/lychnos/es_es/articulos/inteligencia_artificial (último acceso: Agosto de 2019).

Wikipedia. 2019. https://es.wikipedia.org/wiki/Cintur%C3%B3n_de_seguridad (último acceso: Julio de 2019).

—. 2019. https://es.wikipedia.org/wiki/Bus_CAN (último acceso: Julio de 2019).

—. 2019. https://es.wikipedia.org/wiki/Inteligencia_artificial (último acceso: Julio de 2019).

—. 2019. https://es.wikipedia.org/wiki/Sensor (último acceso: Agosto de 2019).

—. 2019. https://es.wikipedia.org/wiki/ECall (último acceso: Agosto de 2019).

—. 2019. https://es.wikipedia.org/wiki/Red_neuronal_artificial (último acceso: Julio de 2019).

—. 2019. https://es.wikipedia.org/wiki/Microcontrolador_PIC (último acceso: Septiembre de 2019).

—. 2019. https://es.wikipedia.org/wiki/System_on_a_chip (último acceso: Septiembre de 2019).

2.12 Definiciones y abreviaturas

- **CAN:** Controller Area Network.

Es un protocolo de comunicaciones desarrollado por la firma alemana Bosch, basado en una topología bus para la transmisión de mensajes en entornos distribuidos. Además, ofrece una solución para la gestión de cableado ya que solo utiliza dos cables.

- **RNA:** Red Neuronal Artificial.

Es un modelo computacional que intenta parecerse al cerebro humano en cuanto a su funcionamiento con neuronas. En la red, el símil de las neuronas biológicas son las neuronas artificiales que, al igual que en nuestro cerebro, se interconectan entre ellas para pasar información. Estas redes se utilizan para resolver problemas en los que no está clara cuál es la relación entre la entrada y el resultado.

- **DAQ:** Data Acquisition; Adquisición de Datos.

Consiste en la toma de muestras del mundo real para generar datos que puedan ser manipulados por un ordenador u otros dispositivos electrónicos.

- **SPI:** Serial Peripheral Interface, Interfaz serie para periféricos.

Se trata de una comunicación serie síncrona utilizada principalmente en sistemas embebidos para distancias cortas de comunicación. Este tipo de comunicación se lleva a cabo de forma full-duplex utilizando la arquitectura de maestro-esclavo con un único maestro.

- **PIC:** Peripheral Interface Controller, Controlador de interfaz periférica.

Los PIC son una familia de microprocesadores de tipo RISC (set de instrucciones reducido) fabricados por Microchip Technology Inc y derivados del PIC1650. Poseen una arquitectura minimalista. (Wikipedia 2019)

- **SoC:** System on Chip, Sistema en chip

Consiste en un circuito integrado que incorpora todos los módulos necesarios para una aplicación. (Wikipedia 2019)

2.13 Análisis de soluciones: Especificaciones de sistema eCall mejorado para vehículos a proponer

Se plantea estudiar sobre un sistema capaz de realizar las mismas funciones que el sistema eCall, ya comentado, y otras nuevas que se van a comentar más adelante. El sistema sería integrado dentro del vehículo por parte de la empresa automovilística que lo desease protegiéndolo al máximo para evitar su destrucción en caso de colisión.

Este sistema deberá ser capaz de:

- Leer todos los sensores necesarios del automóvil que se contemplen más otros que se instalarán, en caso de ser necesario, para poder realizar las posteriores operaciones.
- Crear un registro temporal de todos los datos que obtiene de los sensores leídos.
- Decidir qué tipo de colisión se ha producido. Entendiendo esto como que se crearan diferentes niveles de colisión y este dispositivo debe ser capaz de clasificar el tipo de suceso en uno de esos niveles.
- Realizar diferentes acciones dependiendo el tipo de suceso ocurrido

- Saber el número de ocupantes que se encuentran en el vehículo antes y después del accidente.
- Saber si, por lo menos el conductor del vehículo tiene las constantes vitales en orden o no.
- Obtener la posición GPS actual del vehículo.
- Ponerse en contacto con los servicios de emergencia
- Emitir radiofrecuencias para poder avisar a otros vehículos de que estamos accidentados

2.13.1 Detección y clasificación del accidente

Para que este sistema de emergencia pueda realizar cualquier acción de seguridad lo primordial será detectar que, realmente, ha ocurrido un accidente y, además, saber si ese accidente es la magnitud suficiente para desencadenar todas las acciones del sistema de emergencia. Para ello deberemos encontrar la relación que existe entre el estado del vehículo y un tipo de accidente o un funcionamiento normal de este.

Figura 4: Ejemplo de vehículo gravemente accidentado

Figura 5: Ejemplo de vehículo levemente accidentado

Figura 6: Vehículo en perfectas condiciones

El primer problema al que nos enfrentamos es el de cómo obtenemos este estado del vehículo. Necesitaremos obtener variables físicas, como, por ejemplo, la velocidad, la inclinación, estado de airbags de un coche entre otros parámetros.

Para lograr esto disponemos de sensores. Los sensores son dispositivos capaces de detectar acciones o estímulos externos y responder en consecuencia a estos. En la industria, es un objeto capaz de variar una propiedad ante magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas con un transductor en variables eléctricas. (Pérez Porto y Gardey 2010) (Wikipedia 2019)

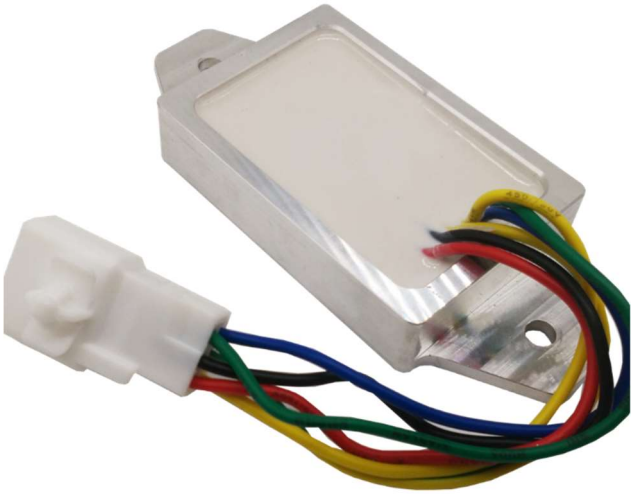


Figura 7: Sensor de inclinación para vehículos SS315 con conexión para bus CAN

Gracias a estos sensores conseguiremos que todas las magnitudes físicas que queremos leer estén disponibles en variables eléctricas. Gracias al desarrollo e innovación de las empresas de vehículos actuales, los vehículos de hoy en día están plagados de sensores los cuales podremos utilizar para obtener los parámetros que nuestro dispositivo necesite. Además, estos sensores están perfectamente preparados para las duras condiciones que hay dentro de un coche como son las vibraciones, las altas temperaturas, sobre todo cerca del motor, y las salpicaduras de agua entre otras, por lo que los hace idóneos para nuestro sistema.

A pesar de que los coches presentan ya instalados de fábrica muchos sensores para todo tipo de variables físicas, cabe la posibilidad que algún coche en el que queramos instalar este sistema de seguridad, éste desprovisto de alguno de los sensores que nuestro sistema debe utilizar para realizar sus acciones. En este caso, deberemos instalar nosotros mismos los sensores necesarios y protegerlos para asegurar su integridad.

Teniendo todos los sensores necesarios ya instalados en el vehículo nos surge el siguiente problema; debemos conectar todos esos sensores a un sistema que lea, clasifique y guarde todas esas señales eléctricas para su posterior tratamiento.

La primera solución que podríamos dar a este sistema es conectar individualmente cada sensor al sistema de adquisición de datos que utilicemos. Esta es una buena solución ya que obtendríamos prácticamente al instante la lectura de los sensores. La contraparte de esta topología es que aumentamos en gran medida el número de cables en nuestro coche ya que necesitamos un cable que interconecte el sensor con el DAQ más otros dos cables de alimentación para el propio sensor y, estando en un medio tan agresivo, es mejor evitarlos en lo máximo posible.

La solución óptima para nuestro sistema es utilizar la solución originaria de Bosh que patentó en los años ochenta y que a día de hoy es la topología más utilizada en vehículos para intercomunicar sensores y circuitos de control de los diferentes sistemas dentro éste. Estamos hablando del bus CAN. Con este protocolo de comunicaciones logra reducir el cableado a un único cable de par trenzado que interconecta todos los sensores y al que se conecta el DAQ para leerlos todos. No es tan rápido como la solución anterior, pero presenta innumerables ventajas que lo hacen el óptimo para utilizarlo en este entorno como son, por ejemplo, gran resistencia a interferencias, comunicación robusta, fácil instalación y, como se ha comentado, muy poco cableado entre otros.

Toda la información procedente de estos sensores será leída a un intervalo regular y todos los datos que se obtengan en cada iteración serán guardados en la memoria del sistema de control. Esta información le permitirá tomar decisiones e incluso la posibilidad de generar un informe con las diferentes lecturas que se han hecho de los sensores, para informar a las emergencias de cara a que planifiquen su intervención (por ejemplo, la necesidad de excarcelación de los ocupantes) y/o para formar parte de un "log" del que poder deducir el motivo del accidente, entre otros.

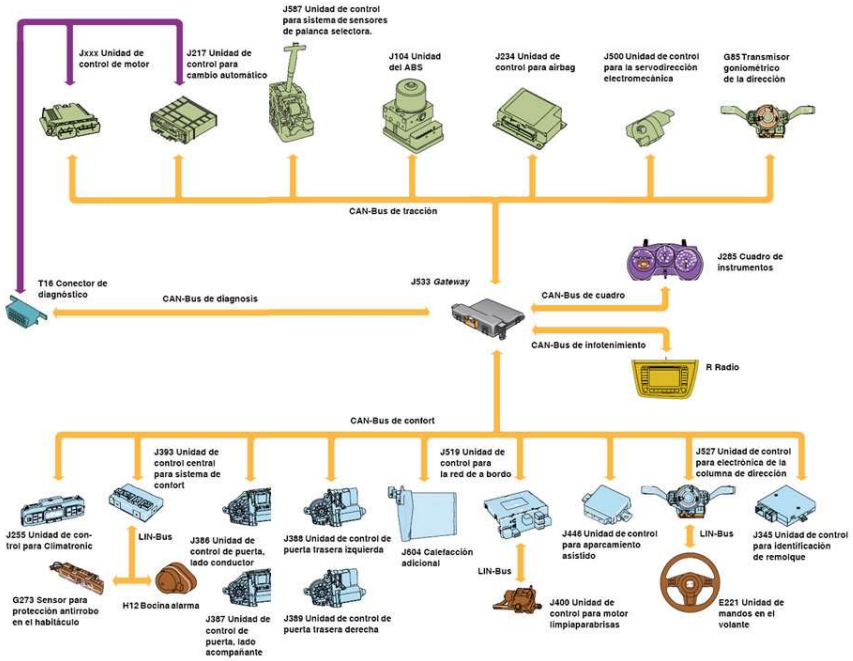


Figura 8: Ejemplo de topología bus CAN de un automóvil genérico

El tercer problema es el de encontrar la relación que hay entre los parámetros obtenidos de los sensores y un tipo de accidente o un funcionamiento normal del vehículo.

La primera solución que podemos adoptar es la de realizar un modelo matemático que logre esta relación. La ventaja de utilizar esta solución es que obtendremos un modelo que será muy fiable en sus resultados y muy rápido en la obtención del resultado. Por la otra parte tenemos las desventajas. A pesar de las ventajas mencionadas este modelo matemático en esta situación sería extremadamente complejo de obtener además de que cabe la posibilidad de que no obtengamos un modelo demasiado fiable ya que deberemos utilizar aproximaciones que darán como resultado variaciones indeseadas en la solución del problema o incluso, no llegar a obtener un modelo matemático que realice esta relación. Otra desventaja más es la de que necesitaríamos realizar un modelo matemático por cada vehículo en el que queramos instalar este dispositivo y aun estando en el mismo vehículo, si cambiáramos uno de los sensores o alguna construcción del vehículo que afecte a este sistema, el modelo matemático que hayamos creado sería aún más impreciso si no lo recalculamos. Por lo que esta solución se hace inviable para nuestro sistema.

La solución óptima sería la de utilizar Deep Learning y Redes Neuronales Artificiales. Con esta herramienta conseguiremos crear un sistema muy versátil y muy fiable gracias a su funcionamiento. Con ella conseguiremos que la toma de decisiones sea muy precisa, independientemente del vehículo, si este lleva instalados los sensores que requiere el sistema. En caso de modificación grande del vehículo, únicamente sería necesario reentrenar la red para que se adapte a estos cambios, pero nunca será necesario cambiar la RNA al completo.

Esta RNA debe poder diferenciar entre tres tipos de accidente más el de funcionamiento no-accidente o estado normal. Estos serán, de menor a mayor gravedad, funcionamiento normal, el coche está en perfectas condiciones de funcionamiento y no se ha producido ninguna colisión; Accidente leve, cuando se ha producido una colisión a baja velocidad en la que ni siquiera se han disparado los airbags del vehículo, o que se ha disparado, pero con una mínima repercusión; Accidente moderado, suceso en el que ya el sistema de seguridad del vehículo establecerá contacto con el conductor. Esta comunicación se realizará por medio de voz con un altavoz y un audio pregrabado que se reproducirá cuando sea necesario. Si este no responde o no interrumpe intencionalmente el lanzamiento automático del sistema de emergencia, se avisará a los servicios de emergencia y el cuarto nivel sería el de accidente grave, en este caso se avisará directamente a los servicios de emergencia.

2.13.2 Avisos y alertas de seguridad y ubicación del accidente

Superada la etapa de adquisición de datos, el guardado de estos y toma de decisiones, tiene lugar la etapa de los avisos de emergencia. Estos avisos se producirán dependiendo del resultado de la RNA la cual será capaz de clasificar el suceso en 4 niveles.

Se plantean dos alternativas: la primera es el modo automático en el que el sistema, sin ayuda humana, será capaz de realizar todas las acciones de emergencia pertinentes. La segunda, el modo manual, en el que es el propio usuario el que se encargará de desencadenar todo el protocolo de emergencia que lleva incorporado este sistema.

2.13.2.1 Modo automático

Automáticamente se deberán desencadenar unas acciones u otras dependiendo del resultado de la RNA. Así pues, podremos diferenciar entre 4 acciones en función de la clasificación efectuada por el sistema:

- **Funcionamiento normal**

Este es el resultado de la RNA el cual debería ser el más común ya que la mayor parte del tiempo de uso de un vehículo es permanecer en el estado de funcionamiento normal. En este caso el coche no ha recibido ningún impacto y todos los sensores están en perfecto estado o la colisión ha sido tan insignificante que los sensores apenas han registrado cambio, por lo que el vehículo estaría en perfectas condiciones, al igual que sus ocupantes. En esta situación el sistema de emergencia no se activaría (al menos no de forma automática, aunque siempre podrá intencionadamente solicitarse de forma manual), y no se daría ninguna alarma ya que es innecesario.

- **Accidente leve**

En este caso, la RNA ha detectado que se ha producido un accidente leve, por ejemplo, el producido a baja velocidad; los airbags no se han disparado y el resto de los sensores habrán registrado un cambio notable, pero lo suficientemente pequeño como para considerar que se ha producido una colisión de baja repercusión. Debido que este tipo de suceso, el accidente es considerado como leve, ya que existe una gran posibilidad de que los ocupantes estén en perfecto estado o suficientemente capacitados para que puedan solicitar ayuda intencionadamente de forma manual si lo necesitan, o continuar la marcha sin mayores consecuencias, si esto no fuera posible. En este caso, nuestro sistema de emergencia tampoco realizará ninguna acción de forma automática.

Aunque no se hayan realizado acciones de emergencia, este sistema crea un registro de los últimos datos tomados para guardar un pequeño histórico de lo sucedido, a modo de “caja negra”. Este informe será guardado en una memoria no volátil para luego poder acceder a ellos y, en el caso que se requiera, realizar un parte de accidente para ayudar a aclarar la responsabilidad de la colisión, en el caso de que esté involucrado otro vehículo, por ejemplo. Cabe destacar que estos datos estarán protegidos (encriptados) y podrán ser extraídos

por personal especializado, agentes de Tráfico (Policía, Guardia Civil, etc. o por técnicos autorizados de la empresa desarrolladora del sistema o del vehículo por motivos de confidencialidad y para evitar posibles manipulaciones no autorizadas.

- **Accidente moderado**

Este resultado de la RNA nos indica que se ha producido un accidente potencialmente grave. En esta situación los airbags y el mecanismo de pretensores de los cinturones se habrían disparado. Además de esto, el resto de los sensores habría detectado un gran cambio desde su estado anterior o puede que incluso alguno de ellos se encuentre destruido. Debido a esto, el sistema de emergencia se activará realizando las siguientes acciones:

En primer lugar, el sistema emitirá un mensaje de voz y tratará de establecer contacto con el conductor, a través del sistema de audio del vehículo de un altavoz propio (redundante, por si el sistema de audio hubiese quedado inutilizado). El sistema esperará un tiempo de cortesía a que intencionadamente, el conductor o algún ocupante detenga el sistema de emergencia, significando que la colisión no necesita los servicios de emergencia. En caso contrario, el sistema intentará enlazar con un dispositivo similar a una pulsera de actividad, que permita conocer algunas constantes vitales del usuario. Como, por ejemplo, el pulso, para conocer su estado, y mediante pulsadores, podremos saber si el conductor acepta o rechaza la emergencia dependiendo de si pulse el botón SOS para aceptarla o el de cancelación para cancelarla.

Para conocer el estado del conductor podemos barajar distintas alternativas. Una de ellas podría ser el instalar en el salpicadero una cámara con reconocimiento facial y con reconocimiento de distintos síntomas que puede sufrir el conductor. Este sistema presenta la ventaja de que el usuario no necesitaría portar nada encima para reconocer algunos de los signos vitales, pero presenta diversas desventajas. La primera es la complejidad de este dispositivo. Para crearlo será imprescindible entrenar otro sistema que permita reconocer lo que nosotros necesitamos. La segunda es que debido a que es probable que nos encontremos con un escenario en el que el vehículo ha resultado muy deteriorado, es posible que este reconocimiento no sirva de nada, ya que será fácil que la cámara se destruya o que el conductor no esté en la posición óptima para que esta cámara cumpla su cometido. La tercera y la más importante es que el sistema que creemos será poco efectivo en sus mediaciones, ya que estimar las constantes de una persona sólo mediante su observación es demasiado complejo.

Figura 9: Ejemplo de puntos que una RNA de reconocimiento facial tomaría sobre el usuario

La solución óptima para este problema es que el usuario porte consigo algún dispositivo que sea capaz de, por lo menos, tomar el pulso del usuario. Por suerte, hoy en día existen multitud de pulseras que son capaces de realizar esta función con bastante precisión y, además, es muy común que cualquier persona porte una consigo mismo que realice otras funciones a parte de esta. Esta pulsera enviaría por conexión inalámbrica los parámetros a nuestro dispositivo de seguridad y este se encargará de analizarlos.

Figura 10: Ejemplo de pulsera que incluye medición de pulso y que se conecta por bluetooth con un dispositivo móvil

El segundo paso que realiza este sistema es el aviso a los coches circundantes de que ha ocurrido un accidente cerca, para evitar una posible segunda colisión con el vehículo que ya está accidentado. Para ello, el sistema contará con una emisora de RF que lanzará una señal con una potencia suficiente para cubrir algunos km de distancia, una frecuencia determinada y normalizada, en la que se trasmitirá información relacionada con el vehículo

accidentado y que podrá ser recibida por sistemas compatibles que porten los vehículos circundantes para estar prevenidos del suceso, su localización; y en caso de ser necesario, detenerse a ayudar al vehículo accidentado.

La tercera acción que realizar es la de crear un pequeño informe con lo sucedido. Este informe deberá incluir las últimas muestras tomadas de los sensores del vehículo, la posición GPS actual del vehículo, así como algunas anteriores, por si el sistema GPS habría quedado destruido, los ocupantes que están en el vehículo actualmente y los que estaban hace unos instantes, por si acaso alguno de ellos habría salido despedido en el impacto y las constantes vitales de, por lo menos, el conductor, las cuales serán provistas por la pulsera que se ha comentado antes.

El último paso y el más importante es el de avisar a los servicios de emergencia para que se movilicen hasta el accidente lo antes posible. Nuestro sistema avisará a estos servicios además de enviarles el informe que se ha generado sobre lo ocurrido. Este informe es muy importante ya que, gracias a él, los servicios de emergencia sabrán más acertadamente como actuar y qué tipo de equipos desplegar por lo que obtendremos una ayuda mucho más eficaz.

- **Accidente grave**

Este es el caso más desfavorable de todos y el menos deseable. Si la RNA ha dado este nivel como resultado significa que estamos ante un accidente de una magnitud muy elevada. Las posibilidades de supervivencia son bajas, pero aún existen por lo que se debe actuar con velocidad. En esta situación el sistema de seguridad se activará de inmediato y realizará todas las acciones que se han comentado en el anterior apartado salvo que, en este caso, el primer paso no se realizará. El dispositivo avisará mediante radiofrecuencia a los vehículos circundantes, creará el informe pertinente y llamará directamente a los servicios de emergencia para que puedan realizar su actuación con la mayor celeridad posible. Aún en el supuesto de que se trate de un accidente con víctimas, es muy necesario que se balice la zona, se limpien los restos, etc., para evitar cualquier otro suceso indeseado a raíz de este.

Debido a la gravedad de la situación no sería eficaz ni eficiente el intentar contactar con el conductor ya que existen altas probabilidades de que este se encuentre inconsciente o incluso peor. Es bastante probable que la mayor parte del vehículo haya quedado muy dañada, por lo que es muy importante que el sistema propuesto esté protegido mecánicamente y que, además, cuente con batería auxiliar para seguir operativo durante un tiempo.

2.13.2.2 Modo manual

Este modo realizará las acciones que se han explicado en el apartado de accidente grave solo que, para que se active, debe ser el propio usuario el que presione un botón de SOS que se encontrara en el salpicadero o el volante de su coche, diseñado para que esté lo suficientemente accesible, pero que no sea fácil su activación fortuita.

El coche contactará con él y él deberá apagar el sistema para que los servicios de emergencia se pongan en camino. En caso de pulsar por equivocación, bastará con que apague el dispositivo y ningún sistema de emergencia será activado.

2.13.3 Medidas de integridad del propio sistema

Como se ha ido mencionando a lo largo del documento, el vehículo de por sí ya es un entorno muy hostil. A parte de esto, nuestro sistema va a trabajar en unas condiciones que son extremadamente hostiles al igual que un accidente automovilístico. Si juntamos estas dos condiciones obtenemos como resultado un entorno donde las posibilidades que cualquier aparato mecánico o electrónico resulte inoperativo son altísimas por lo que nuestro sistema debe de ser capaz de soportar todo esto.

Figura 11: Ejemplo de vehículo muy gravemente dañado debido a una colisión

Por suerte, los sensores ya instalados en el vehículo, y los que instalaremos, ya están preparados para soportar las condiciones de temperatura y vibraciones que se producen. Sin embargo, en caso de fuerte colisión es posible que estos se destruyan, pero no habría mucho problema ya que el DAQ tomará hasta la última muestra que estos envíen.

Lo mismo sucede con en bus CAN que utilizaremos para leer todos sensores. Puede que se destruya en el impacto, pero el DAQ ya se habrá encargado de guardar todos los datos anteriores.

Como se puede deducir, todos los sensores y sistemas con los que cuenta el automóvil ya están preparados para soportar las condiciones que se dan dentro de este entorno, por lo que los esfuerzos han de dirigirse hacia la protección de los dispositivos que nosotros instalemos en él. Así pues, como lo más sensible será el microcontrolador que realizará todas las funciones de adquisición de datos y avisos de emergencia, con todos los componentes externos, deberemos alojarlos en una especie de “caja negra”, como la presente en los aviones, con las características necesarias para absorber fuertes impactos que logren que el sistema siga funcional y operativo tras grandes impactos.

Esta caja debe ser capaz de proteger nuestro dispositivo ante las condiciones de temperatura, vibraciones, etc., durante el funcionamiento normal del vehículo. Además, debe ser capaz de aguantar una colisión de automóvil sin destruirse, para así ser capaz de extraer los datos que lleva consigo y de articular la petición de ayuda.

Figura 12: Ejemplo de caja negra de un avión

2.13.4 Esquema general de funcionamiento

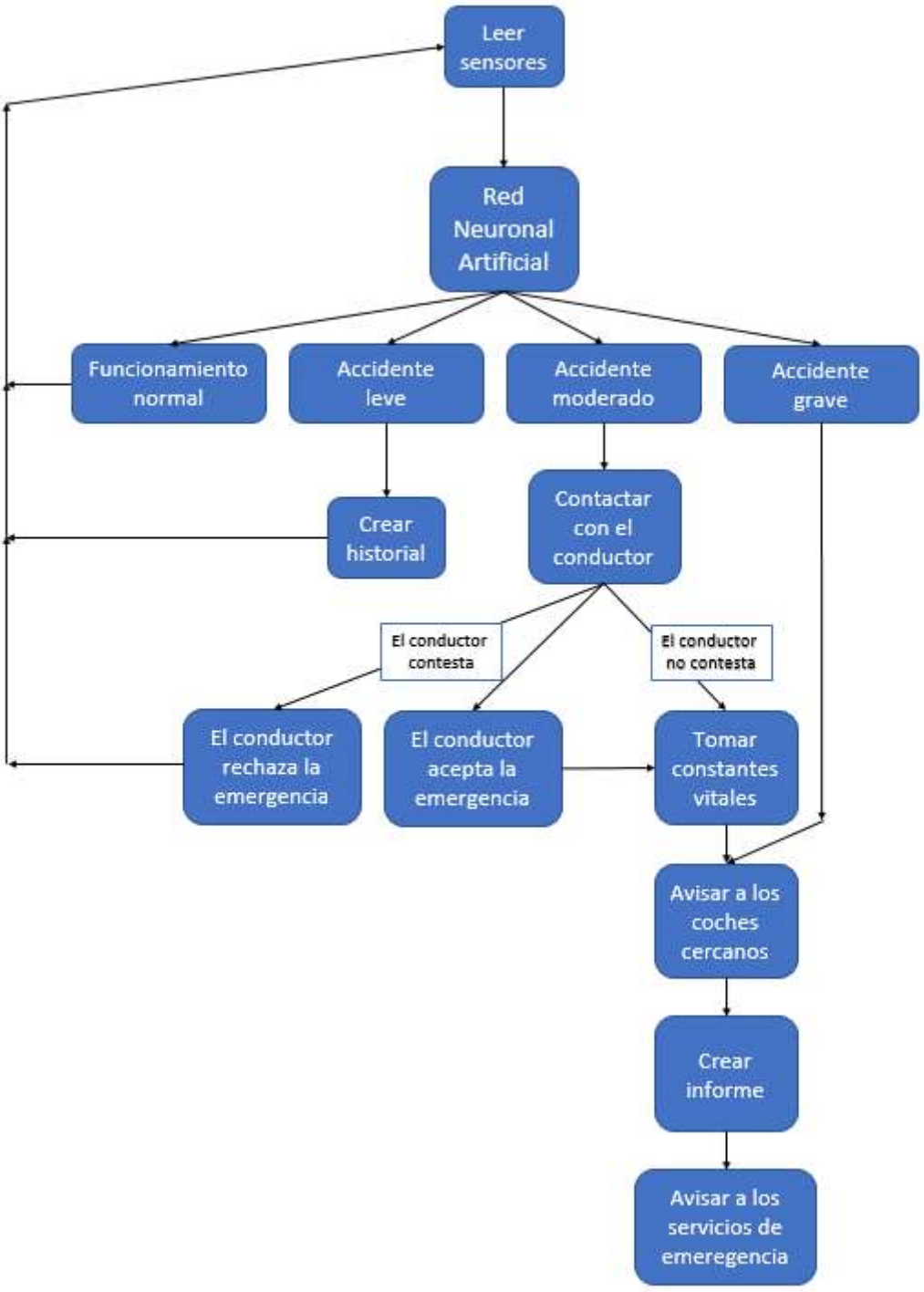


Figura 13: Diagrama general de funcionamiento

El sistema desde el inicio estará tomando datos de todos los sensores. Todos esos datos serán guardados en la memoria volátil del microprocesador para llevar un registro de todo lo que está pasando. Algunos de estos datos sufrirán un pretratamiento para ser enviados a la red neuronal. Una vez realizado esto, estos datos serán los que alimenten la capa de entrada de la red neuronal

artificial. Esta red nos dará el resultado en forma de porcentajes. El porcentaje más alto será el resultado que deberemos utilizar. En caso de que dos neuronas den un resultado elevado, se realizará el protocolo de la de mayor nivel

En el caso de que el resultado sea “Funcionamiento normal” que debería ser el más común, no se realizará ninguna acción. El microprocesador al obtener este resultado continuara leyendo los sensores del vehículo para la siguiente iteración.

Si obtenemos como resultado “Accidente leve” el microprocesador creará un informe con los últimos datos guardados en su memoria volátil y lo guardará en una memoria no volátil para, en caso de ser necesario, utilizar estos datos guardados.

En el caso de obtener como resultado “Accidente grave”, se contactará con el conductor por medio de un altavoz. Si este pulsa el botón de SOS significará que sí que necesita que se active el protocolo de emergencia, y, al contrario, en caso de que pulse el botón de cancelar, querrá decir que no necesita ningún servicio y que puede requerirlo por el mismo. Cabe destacar que, si el conductor no contesta, el sistema se activará automáticamente al paso de unos segundos porque significaría que no es capaz de aceptar el protocolo ni de rechazarlo.

Si el conductor rechaza la emergencia, el sistema volverá a leer los sensores al igual que si no hubiera ocurrido nada. Sin embargo, si se ha desencadenado este protocolo, el microprocesador tomará las constantes vitales del conductor, avisará de que se ha producido un accidente cerca de ellos mediante radiofrecuencia para que tengan cuidado, creará un informe con los datos de los sensores y lo guardará en la memoria no volátil y enviará este informe a los servicios de emergencias para que se atiendan la emergencia lo antes posible.

Por último, si la red neuronal da como resultado “Accidente grave” el microprocesador se pondrá de inmediato a trabajar en atender la urgencia lo antes posible. Como se ve en el dibujo, directamente se pasaría a avisar a los vehículos que se aproximan, se crearía el informe y se enviaría a los servicios de emergencia para actuar con la mayor celeridad posible.

Cabe destacar que, aunque esto será un proceso automático, el usuario en cualquier momento podrá pulsar el botón SOS, iniciándose así el protocolo que se ha comentado en el párrafo anterior, ya que no sería necesario comunicarnos con él debido a que es él mismo el que ha solicitado que se desencadene el protocolo de emergencia.

2.14 Esquema general de bloques del sistema propuesto

Se presenta a continuación un diagrama de bloques en el que se muestran los principales elementos que constituyen el sistema propuesto.

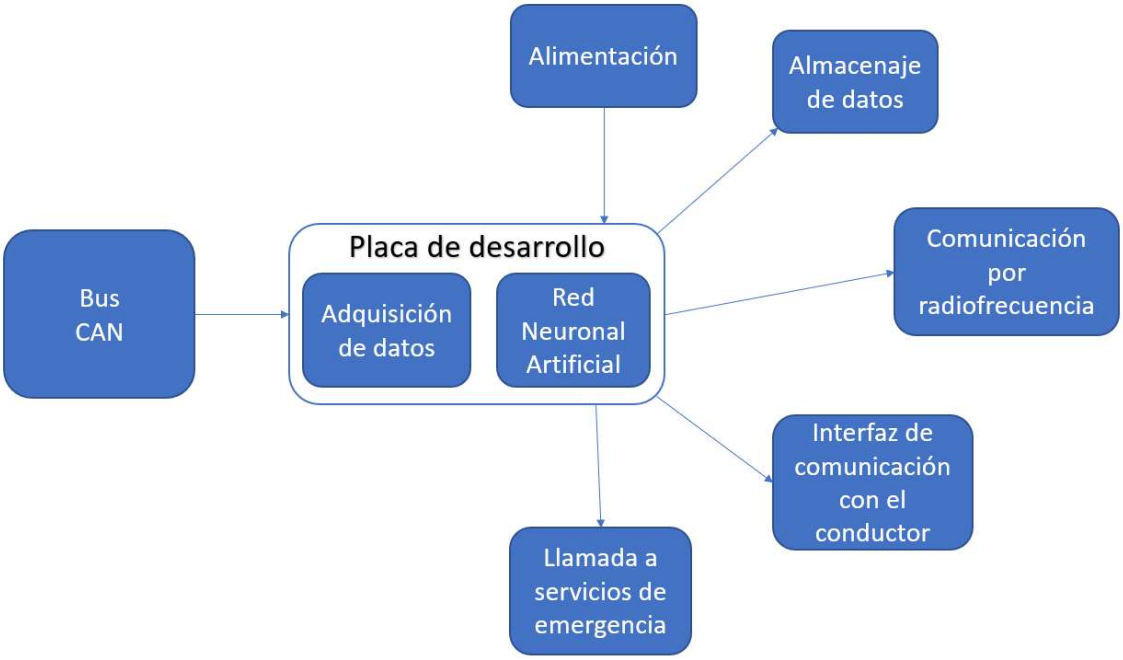


Figura 14: Esquema general de bloques

El sistema que se ha propuesto se puede dividir en los bloques que se han dibujado arriba.

En la parte central se encuentra la placa de desarrollo que será la encargada de gestionar todo el sistema que se plantea. Dentro de esta placa nos encontramos el bloque de adquisición de datos y la red neuronal artificial. El primero se encarga de adquirir los datos de los sensores que viajan por el can bus y el segundo se encarga de tomar una decisión de acuerdo con los datos de los sensores obtenidos.

En el bloque “Bus CAN” se explicará todo lo relacionado con el bus CAN del coche ya que todas las comunicaciones entre sensores se realizan mediante ese protocolo. Aquí se incluyen los sensores necesarios que debe poseer el vehículo o los que se deben instalar para que el sistema funcione correctamente.

El bloque “Alimentación” hace referencia a los sistemas que se ocupan de proporcionar la alimentación nuestro sistema y qué extras se deben añadir para evitar posibles cortes de suministro cuando se produzca una colisión.

El bloque “Almacenaje de datos” contiene los recursos necesarios que posibilitan la salvaguarda de los informes que crea el sistema cuando se produzca colisión. Así mismo, también se muestra el módulo necesario para escribir los datos en ese medio. En nuestro caso, ese medio podría ser una tarjeta microSD.

El bloque “Comunicación por radiofrecuencia” contiene los sistemas necesarios para realizar la comunicación entre los vehículos y generar el aviso del suceso. Será una comunicación mediante radiofrecuencias y se llevarán a cabo con módulos periféricos específicos.

El bloque “Interfaz de comunicación con el conductor” se encarga de materializar esta interacción con el conductor. Se presentan dos tipos de comunicación, una directa y otra indirecta. Esta última necesaria en caso de que el conductor se encuentre en un estado de inconsciencia o que no pueda, por sus propios medios, interactuar directamente con el usuario.

Por último, el bloque “Llamada a servicios de emergencia” está formado por los módulos necesarios para establecer comunicación con el 112 y solicitar la asistencia.

A continuación, se analizan y describen en detalle cada uno de estos bloques que constituyen el sistema.

2.15 Bloque bus CAN

Uno de los bloques descritos debe ocuparse necesariamente de la comunicación del bloque placa de desarrollo con los sensores y elementos de control ya instalados en el vehículo. El bus de comunicaciones utilizado como estándar en los automóviles actuales, y que por tanto está implementado en los vehículos en varios anillos: confort, seguridad, gestión del motor, etc., es el bus CAN.

2.15.1 Origen

El bus CAN surgió por la necesidad de interconectar los diferentes componentes del automóvil reduciendo el cableado necesario para ello. Comenzó en 1983 gracias a la empresa Robert Bosch GmbH la cual posteriormente publicó varias versiones de la especificación CAN.

Los primeros fabricantes en producir controladores CAN fueron Intel y Philips.

En el año 1993 se publicó el estándar ISO 11898 del bus CAN y ha sido desde entonces un estándar para la organización internacional para la normalización.

2.15.2 Definición

CAN (Controller Area Network) es un protocolo de comunicaciones basado en la tipología bus para la transmisión de mensajes. También, ofrece una solución para la gestión de la comunicación entre varias CPUs.

Este protocolo nos ofrece los siguientes beneficios

- Gran inmunidad a las interferencias
- Habilidad para el autodiagnóstico
- Corrección de errores
- Como es un protocolo normalizado, economiza la tarea de comunicar subsistemas de diferentes fabricantes sobre una red o bus común
- El anfitrión delega la carga de comunicaciones a un periférico inteligente, así este puede realizar sus tareas con un mayor margen de tiempo
- Es una red multiplexada, por lo que se reduce considerablemente el cableado

2.15.3 Principales características de CAN

CAN se basa en el paradigma de comunicación que describe una relación entre un productor y uno o varios consumidores. CAN es un protocolo orientado a mensajes, a cada uno de ellos se le asigna un identificador y el conjunto de mensaje más identificador se le denomina trama. Al estar todos los dispositivos conectados a la misma red, todos ellos reciben el mensaje y gracias a este identificador, los nodos de esta red deciden si aceptar o rechazar el mensaje. Sus características principales son:

- Prioridad de mensajes
- Garantía de tiempos de latencia
- Flexibilidad en la configuración
- Recepción por multidifusión con sincronización de tiempos
- Sistema robusto en cuanto a consistencia de datos
- Sistema multimaestro
- Detección y señalización de errores
- Retransmisión automática de tramas erróneas
- Distinción entre errores temporales y fallos permanentes de los nodos de la red y desconexión autónoma de nodos defectuosos

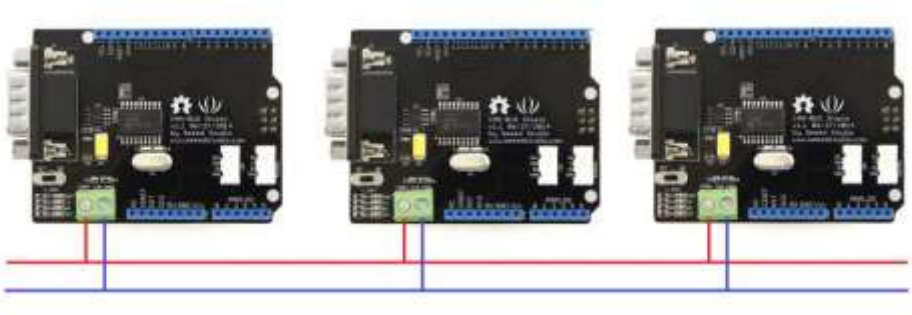


Figura 15: Esquema de una red CAN con tres nodos

2.15.4 Tipos de bus CAN

2.15.4.1 CAN de alta velocidad

Está recogido en la ISO 11898-2. Utiliza un único bus lineal al que se le conectan los distintos dispositivos. En los extremos presenta resistencias de 120Ω para evitar posibles reflexiones de la señal que puedan perturbar la comunicación. Con este tipo de red se alcanzan velocidades de hasta 1Mbit/s.

2.15.4.2 CAN de baja velocidad tolerante a fallos

Está recogido en la ISO 11898-3. Este CAN puede utilizar un bus lineal, en estrella o múltiples buses en estrella conectado por un bus lineal. Cada nodo de este bus debe presentar una resistencia de terminación no inferior a los 100Ω. Según la norma, se permiten velocidades de hasta 125 kbits/s.

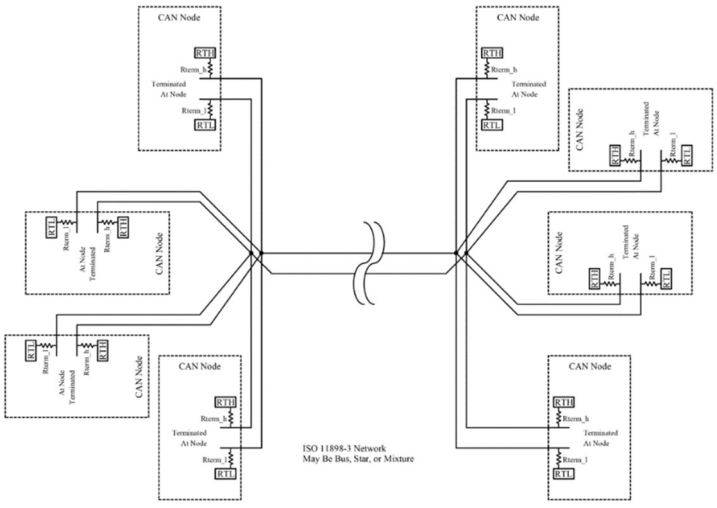


Figura 16: Esquema de una red CAN de baja velocidad con configuración de varios buses estrella con un bus lineal de interconexión

2.15.4.3 CAN FD (flexible data-rate)

En 2012, Bosch lanzó el CAN FD el cual aumentaba significativamente la tasa de transferencia después del arbitraje. Es posible transmitir mayor cantidad de datos en el mismo tiempo y la frecuencia se puede multiplicar hasta por 8. De momento solo está definido la capa de enlace de datos. Las especificaciones están recogidas en el borrador ISO/DIS 11898-1 2015.

2.15.5 Capa física

La transmisión de señales se lleva a cabo a través de dos cables trenzados denominados CAN_H y CAN_L. Este bus presenta dos estados, el estado dominante, que significa que entre estos dos cables existe una diferencia de potencial de al menos 1.5v (la tensión debe estar entre 1.5 y 3v según la norma) y el estado recesivo que significa que entre los dos cables no existe diferencia de potencial. Gracias a este tipo de transmisión diferencial, conseguimos una mejor protección ante interferencias electromagnéticas.

El estándar CAN no especifica un conector específico para el bus, pero existen varios formatos comúnmente aceptados como el conector D-sub de 9 pines con las señales CAN_H y CAN_L en los pines 7 y 2 respectivamente

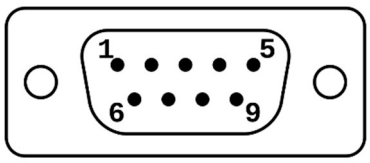


Figura 17: Conector D-sub de 9 pines

Todos los nodos de una misma red CAN deben trabajar a la misma frecuencia y, como este bus no presenta ninguna señal de reloj común para todos los nodos y cada uno utiliza la suya propia, se debe establecer un método de sincronización de bits. Esto se hace especialmente importante cuando estamos en la fase de arbitraje porque cada nodo debe saber lo que envía el y el resto de los nodos. Esta sincronización presenta las siguientes fases:

- Segmento de sincronización: Es la primera parte del tiempo nominal del bit y es la zona donde se supone que va a haber un cambio de recesivo a dominante.
- Segmento de propagación: En este intervalo de tiempo se compensan los posibles retardos que se ocasionen en el bus
- Segmentos de fase 1 y 2: Se usan para llevar a cabo la resincronización de los nodos. Normalmente el nodo 1 es alargado

y el 2 acortado. El punto de muestreo se encuentra justo al terminar el nodo 1

2.15.6 Capa de enlace de datos

Uno de los protocolos de acceso al medio que utiliza este bus es el CSMA/CD + AMP (Carrier Sense Multiple Access / Collision Detection + Arbitration on Message Priority). En este protocolo todos los nodos se conectan al mismo bus y realizan una escucha y así saben cuándo un mensaje está siendo transmitido. De este modo pueden saber cuándo el bus está en estado dominante y evitan transmitir y se limitan a escuchar. Puede darse la situación de que dos nodos intenten transmitir bits diferentes a la vez por lo que en este caso se produciría una colisión de bits. En este caso, el nodo que intentaba transmitir el valor recesivo detecta la colisión y pasa a estado pasivo. Debido a esto, es muy importante que todos los nodos estén perfectamente sincronizados.

El arbitraje se produce durante los primeros bits de una trama, durante la transmisión del identificador. Cuando este proceso finaliza, solo debe haber un nodo con el control del bus que es el que transmitirá la información. Por esto, es muy importante que cada nodo tenga su identificador único dentro de la red. Las tramas con identificador con mayor número de bits dominantes tendrán la mayor prioridad. Observando los identificadores de cada trama se podrá determinar de forma muy precisa el orden de envió de los mensajes.

Si un nodo pierde su arbitraje, este pospondrá los mensajes a transmitir hasta que le vuelva a tocar el turno.

La aceptación o rechazo de los mensajes se produce de la siguiente manera.

1. El nodo lee el mensaje
2. Se realiza una AND lógica entre la máscara del buffer de recepción y el identificador del mensaje
3. Se comprueba si el resultado de la operación anterior coincide con el filtro que se le ha impuesto al nodo
4. Si coincide con el filtro, el mensaje será leído y, en caso contrario, el mensaje será rechazado y no se leerá

2.15.6.1 Formato de trama

- → **Inicio de trama:** 1 bit de longitud. Comienza la transmisión del mensaje. Posee el valor dominante (0)
- → **Identificador (campo de arbitrio):** de 11 a 29 bits. Identificador único que establece la prioridad del mensaje.
- → **Petición de transmisión remota:** 1 bit. Dominante para tramas de datos y recesivo para peticiones remotas
- → **IDE:** 1 bit. Dominante para el formato con identificador de 11 bits

- → **Bit reservado:** 1 bit. Bit reservado que debe ser dominante
- → **Código de longitud de datos (DLC):** 4 bits. Número de bytes de datos en el mensaje con un máximo de 8 bytes
- → **Campo de datos:** de 0 a 64 bits. Datos de la trama
- → **CRC:** 15 bits. Código que indica si los datos fueron correctamente transmitidos
- → **Delimitador de CRC:** 1 bit. Delimita el CRC con nivel recesivo
- → **Hueco de acuse de recibo (ACK):** 1 bit. El transmisor envía recesivo y cualquier receptor envía dominante.
- → **Delimitador de ACK:** 1 bit. Delimita el ACK con nivel recesivo
- → **Fin de la trama (EOF):** 7 bits. Indica el fin de la trama con todos bits a nivel recesivo.

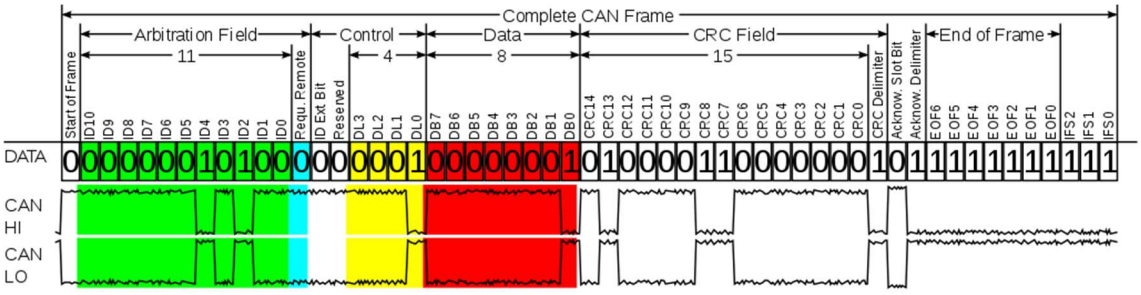


Figura 18: Ejemplo de trama de bus CAN

2.15.6.2 Tipos de formatos

- → **Formato estándar**

Posee un identificador de 11 bits, con el que asigna la prioridad a los diferentes nodos, y puede llegar hasta los 107 bits la trama completa.

- → **Formato extendido**

Posee dos identificadores (A y B) que llegan hasta los 32 bits, incluyendo dos bits, el SRR y el IDE que se sitúan al final del identificador A y al principio del identificador B respectivamente. Esta trama puede llegar hasta los 128 bits. Al ser de gran tamaño, no se utiliza en redes que necesiten un gran desempeño.

2.15.6.3 Tipos de tramas

- → **Trama de datos**

En esta parte se envían los datos que se requiera transmitir por el bus. Puede contener hasta 64 bits de información (8 bytes) en cualquiera de los formatos.

- → **Trama remota**

Puede darse el caso de que, uno de los nodos, requiera información de otro nodo. En este caso, este nodo puede hacer una petición al otro nodo mediante una trama remota. Esta trama se caracteriza por tener el bit RTR a nivel recesivo y no tiene campo de datos. El identificador será el identificador del nodo al que le quiere pedir información. Cuando este nodo la reciba, enviarán los datos pertinentes a través del bus.

→ **Trama de error**

Este tipo de trama no sigue las normas del formato de tramas CAN y se genera y se transmite por el bus cuando un nodo detecta un error en la red. Esto provoca que el resto de los nodos envíen también tramas de error. Existe un mecanismo, dentro de la red, que provoca que un nodo se encargue de bloquear la red enviando constantemente tramas de error.

→ **Overload frame (Trama de saturación)**

Esta trama la genera un nodo que se encuentra realizando demasiadas tareas y no puede transmitir. En este caso, el bus, generará un retardo mayor entre tramas para que el nodo ocupado se libere y transmita. Al igual que la trama de error, esta no respeta las normas de las tramas CAN.

2.15.6.4 Separación entre tramas

Entre trama y trama debe haber como mínimo 3 bits recesivos. En el momento en el que se reciba un bit dominante, se considerará que es el inicio de una trama nueva. La trama de error y la de overload, al no respetar las normas CAN pueden saltarse esta regla.

2.15.6.5 Bits de relleno o bit stuffing

Para garantizar el sincronismo entre los nodos, es necesario añadir un bit de polaridad opuesta cada vez que se dan 5 bits al mismo nivel. Es necesario realizar esto debido a la codificación del protocolo CAN que nunca retorna al nivel de tensión 0. Estos bits intercalados son eliminados por el receptor cuando acepta la trama. (Wikipedia 2019), (Osés 2015, 5 - 14)

2.15.7 El bus CAN en nuestro sistema

En nuestro sistema utilizaremos el bus CAN del vehículo ya que es el medio por el que se interconectan los sensores de nuestro coche. Existen varios anillos de CAN, por lo que nuestro sistema debe ser capaz de conectarse a ellos.

2.15.7.1 Sensores comúnmente instalados en los vehículos

Para que este nuestro sistema pueda llegar a funcionar necesita de una serie de sensores que normalmente están ya instalados en los vehículos, pero, dependiendo del modelo y del año de su construcción es posible que no estén presentes todos. En caso de que alguno de estos no se encuentre instalado, se deberá instalar uno del mismo fabricante desarrollador del coche en cuestión porque ya viene preparado para todas las condiciones que se dan en el vehículo.

Estos sensores son:

- **De guiñada**

Se trata de un sensor de alta tecnología que nos da la velocidad angular en el eje vertical del vehículo en grados o radianes por segundo.

En el caso de que este sensor falte en nuestro coche se deberá instalar uno que ya se instale en otros vehículos de la misma marca ya que la empresa en cuestión se habrá encargado de elegir la mejor opción para estos.

Este deberá ser colocado en la zona más centrada posible del coche ya que debe registrar la velocidad angular en el eje vertical del vehículo.

Figura 19: Ejemplo de sensor de guiñada para vehículos del grupo Volkswagen

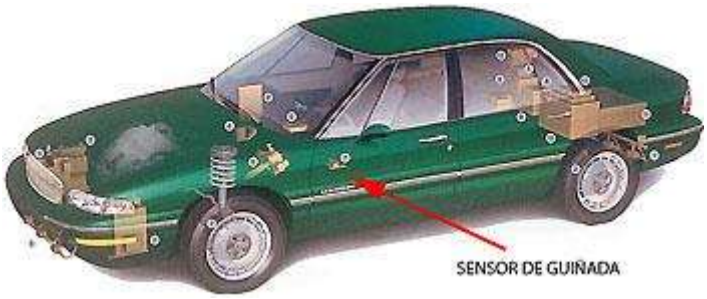


Figura 20: Localización del sensor de guiñada en un coche

- **De inclinación**

Se trata de un sensor que nos da la inclinación actual del vehículo en un eje.

Se deberán instalar dos de estos sensores ya que es necesario medir la inclinación del vehículo en el eje X y Z. La instalación del sensor que detecte la inclinación en el eje X debe estar lo más próximo a este eje. El del eje Z debe estar próximo a este eje. Así, obtendremos medidas más precisas.

Figura 21: Sensor de inclinación de la marca BOSH

- **Acelerómetro**

Este sensor nos indica el cambio de velocidad que está sufriendo el coche en una dirección.

Será necesario instalar dos de estos sensores ya que necesitaremos mediar la velocidad lateral y frontal del vehículo. Existen sensores de este tipo que ya incorporan mediciones para las dos aceleraciones que también podemos instalar.

Figura 22: Sensor de aceleración de la marca BOSH

- **Colisión**

Este sensor nos indica si el vehículo ha sufrido una colisión. Este sensor es muy importante ya que nos va a asegurar que se ha producido tal suceso.

Normalmente se coloca en la parte frontal del coche porque la mayoría de los impactos se producen frontalmente, pero lo idóneo sería repartir más sensores de estos por el resto del vehículo.

Figura 23: Sensor de impacto para vehículos de la marca BMW

- **GPS**

El GPS o Sistema de Posicionamiento Global es una red compuesta por más de 30 satélites que orbitan la tierra y que conjuntamente tienen una visión global de esta. Solo es necesario que 4 satélites estén visibles para nuestro dispositivo para que se conozca la latitud, longitud y altura a la que se encuentra.

Si más satélites forman parte de en este proceso, las medidas serán más exactas.

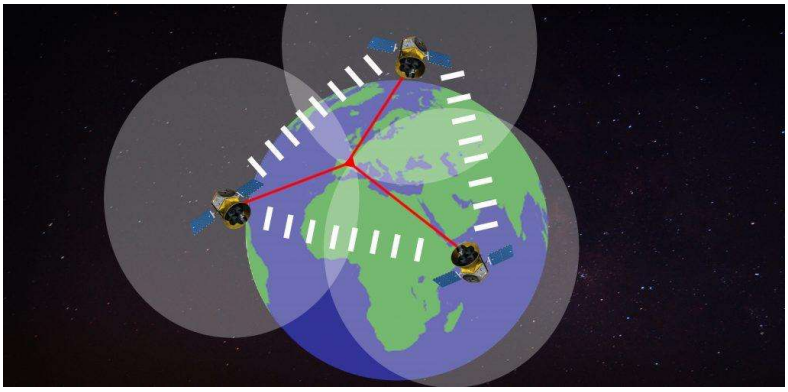


Figura 24: Sistema de geolocalización GPS

Ya son muchos los coches que en la actualidad incorporan GPS para navegación o incluso búsqueda y rescate, pero es muy posible que debamos instalar este sensor si queremos que funcione el sistema de emergencia. En el mercado existen varios modelos; en nuestro caso utilizaremos un dispositivo que irá conectado al puerto OBD del coche en caso de que el vehículo no lo incorpore.

Figura 25: Puerto OBD de un automóvil

Figura 26: Dispositivo GPS GL 500

- **Presión en el asiento**

Este sensor es el encargado de detectar si hay una persona sentada en el asiento o no.

La mayoría de los vehículos ya lo incorporan y se utiliza para avisar al conductor de que no se ha puesto el cinturón. En nuestro caso lo utilizaremos para saber cuántos ocupantes había antes y después del accidente.

Se debe colocar como mínimo en el asiento del conductor y como máximo en todos los asientos. Esto deberá ser configurado para que el microcontrolador sepa cuantos sensores de este tipo tiene.

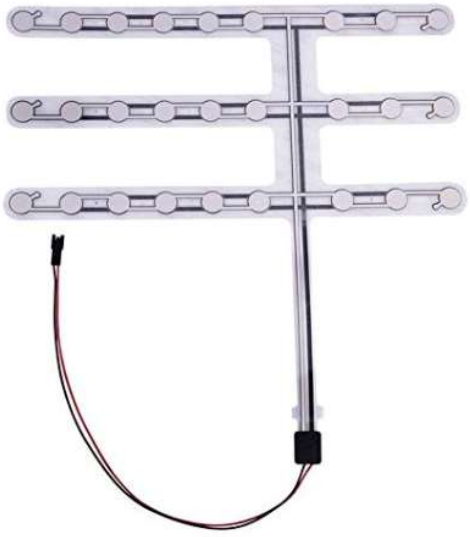


Figura 27: Sensor de presión para colocar en el asiento de un automóvil

- Airbag

Es un sistema de seguridad pasiva que se despliega en diferentes partes del habitáculo del coche para proteger a los ocupantes de golpearse contra la estructura del coche. Generalmente todos los coches poseen al menos uno, en el volante, pero los más modernos incorporan estas bolsas en distintos lugares para proteger a todos los ocupantes.

Figura 28: Interior de in vehículo con los airbags desplegados

El airbag no es un sensor propiamente dicho. Pero cuando este se despliega sí que enviará una trama por el bus CAN que es lo que a nosotros nos interesa.

2.15.7.2 Otros sensores que considerar

Hay otros sensores que podríamos llegar a considerar para poder clasificar aún mejor el accidente. Algunos de estos están, normalmente, instalados en los coches, pero otros no.

- **Sensor de llama**

Este tipo de sensor es capaz de detectar fuego que se produzca cerca de él. No es comúnmente encontrado en los coches, pero sería una buena opción instalar varios de estos por el coche ya que un incendio es un gran indicativo de se está produciendo una situación de emergencia.

- **Sensor de deformación**

Este sensor detecta la deformación que se está produciendo en el medio que ha sido instalado. Hay coches que, sí que presentan varios de estos instalados en su carrocería, pero son la menoría.

La carrocería de los coches actuales esta especialmente diseñada para absorber parte del impacto para así conseguir reducir la fuerza que experimentan los ocupantes. Debido a esto, los sensores de deformación nos darían bastante información sobre lo que le está sucediendo al vehículo. Por lo que la instalación de varios de estos a lo largo de la carrocería de nuestro vehículo sería de gran utilidad para clasificar, aún mejor, el accidente.

- **Frenada**

Este no sería un sensor en sí, sino que se trata de una trama CAN que se enviará por el bus cuando el pedal de freno se accione.

Esta detección de la pisada del pedal puede sernos útil para saber que ocurre justo en los instantes anteriores al accidente, ya que es bastante común accionar el freno cuando nos dirigimos hacia un accidente.

- **Sensor de giro en el volante**

Este sensor nos indicaría la velocidad con la que el conductor está girando el volante. Una velocidad muy grande podría ser indicativo de un volantazo por lo que existen altas probabilidades de que nos encontremos a las puertas de un accidente.

Todos estos sensores estarán sujetos a pruebas por lo que es posible que alguno de estos sensores no será lo suficientemente relevante para utilizarlo en la clasificación del accidente.

Además, deben ser enganchados como nodos de los propios buses CAN que ya implementa el vehículo. Para ello se deberá trabajar con la empresa fabricante del coche en cuestión ya que ella ha sido la responsable tanto de crear el protocolo de ese bus como de unir a él el resto de los sensores y dispositivo que ya implementa. Cada sensor tendrá unas tramas CAN que enviará por el bus, siendo estas las que nosotros deberemos captar con nuestro DAQ. Los que ya estén instalados, tendrán las tramas CAN que envían predefinidas, por lo que el fabricante deberá facilitárnosla.

2.16 Bloque placa de desarrollo

Otro de los bloques propuestos y, se podría decir que es el núcleo de este sistema, es el de la placa de desarrollo. Esta placa debe albergar el sistema de

adquisición de datos y la red neuronal de toma de decisiones. Por lo tanto, esta placa debe ser capaz de conectarse a los buses CAN del coche y tener memoria y capacidad de cómputo suficiente para abarcar la red neuronal que tomará las decisiones pertinentes. Además, debe ser capaz de comunicarse mediante el estándar SPI necesario para enviar datos al módulo de almacenaje de datos.

Existen en el mercado muchas opciones que podemos utilizar que cumplan estas características. Como por ejemplo la Raspberry Pi, una buena opción, diferentes tipos de FPGAs o incluso podríamos diseñar y crear, con ayuda de una FPGA, un SoC que incluya todo lo que necesitamos. Pero, en nuestro caso, nos hemos decidido por un ChipKIT max32 que incluye un microprocesador PIC de 32 bits.

Figura 29: Placa de desarrollo Raspberry Pi 4 modelo B

Se ha elegido esta opción ya que su obtención es muy sencilla obtener además de ser una placa de desarrollo muy barata. Se trata de una placa fácil de utilizar y de la que hay mucha información y tutoriales útiles para su utilización. Además, dispone de infinidad de módulos que podemos conectar a esta para realizar aplicaciones diversas.

ChipKit Max32 es un sistema de desarrollo para uso no profesional desarrollada por Digilent el cual está basado en Arduino.

Figura 30: ChipKit max32

Esta placa de desarrollo es similar a un Arduino MEGA y presenta las siguientes características:

- 512KB de memoria Flash
- 128KB de memoria RAM
- Oscilador de 80MHz
- 4 SPI y 5 conectores I2C
- 16 canales ADC de 10 bits
- 5 salidas PWM
- 2 controladores CAN
- 83 pines de entrada/salida
- 16 salidas analógicas
- Conexión mini-B
- USB 2.0
- 10/100 Ethernet MAC
- LED programable on-board
- 3.3v de alimentación

(Digilent 2015)

Además, esta placa tiene forma similar a la del Arduino MEGA por lo que es compatible con la mayoría de “shields” de protección de ésta para asegurar aún más su integridad.

2.16.1 Adquisición de datos

Este ChipKIT posee la capacidad de conectarse a dos buses distintos por lo que el coche en el que se instale deberá tener como máximo dos o, en su defecto, que los sensores que necesitamos van a estar solo en dos de ellos.

Como nuestra placa va a actuar como DAQ, únicamente será necesario que dentro del bus se posicione como nodo de escucha ya que no necesita

enviar nada. Se limitará a aceptar las tramas CAN que nosotros configuremos dependiendo del vehículo y los de sensores que utilicemos ya que los coches de cada fabricante tienen sus propias tramas CAN.

Esta configuración se lleva a cabo vía software con el software Arduino IDE. Se configurará el nodo de tal forma que acepte las tramas que nos hacen falta y rechace el resto. Además, el propio fabricante de esta placa nos proporciona una librería y un ejemplo que podremos usar para llevar a cabo nuestro cometido de forma más rápida.

La lectura del bus se realizará a intervalos de tiempo equivalentes y se guardarán en la memoria volátil del microchip las muestras tomadas en los últimos 5 minutos, para que, en caso de accidente, todas estas muestras se guarden en una memoria no volátil para su posterior extracción. El intervalo de tiempo de muestreo lo podrá configurar el fabricante y como máximo debería tener un tiempo de un segundo. Si se programase un tiempo mayor, es posible que perdamos información valiosa ya que un accidente se produce en escasos segundos.

2.16.2 Procesamiento de datos y toma de decisiones

La siguiente parte del módulo de placa de desarrollo es la Red Neuronal. Esta se debe encargar de tomar una decisión de acuerdo con lo que esté ocurriendo en el coche. A continuación, se explica lo relacionado con inteligencia artificial, Deep Learning y redes neuronales artificiales.

2.16.2.1 Inteligencia artificial

2.16.2.1.1 Definición

La inteligencia artificial es una rama de la informática, con fuertes influencias en otros campos tales como la lógica y la ciencia cognitiva, que tiene como objetivo hacer que las máquinas imiten las funciones cognitivas de los seres humanos como son, por ejemplo: percibir, razonar, aprender y resolver problemas. Una máquina inteligente sería un sistema capaz de leer y reconocer su entorno, tanto interpretando los datos obtenidos como aprendiendo de ellos y por último, resolviendo el problema que se le asigne a través de la adaptación flexible.

Estas son características porque a igualdad de condiciones de entrada obtienen los mismos resultados. Esto presenta sus limitaciones por lo que con el paso de los años evolucionó a Machine Learning y posteriormente a Deep Learning que se explicarán más adelante. (Olmo 2016)

2.16.2.1.2 Historia

Esta rama nació en una reunión celebrada en el verano de 1965 en Dartmouth (EE. UU.) en la cual participaron J. McCarthy, M. Minisky, N. Rochester y C. E. Shannon. Estos presentaron una propuesta en la que aparece por primera vez el término de "Inteligencia Artificial". Se llevaba 5 años trabajando en este campo y se presentaron varias propuestas, pero no fue hasta ese momento hasta el que fue aceptado por la comunidad investigadora.

A pesar de haber aceptado ese término en el año 1965, mucho antes ya se habló sobre este tema. Por ejemplo, Ramón Llull en 1315, tuvo la idea de que el razonamiento podía ser llevado a cabo de manera artificial y en 1936, Alan Turing, desarrolló la máquina universal que demuestra lo viable que es algo físico para implementar cómputos formalmente definidos. Warren McCulloch y Walter Pitts en 1943 presentaron su modelo de neuronas artificiales.

2.16.2.1.3 Tipos

- **Sistemas que piensan como humanos**

Son sistemas que tratan de simular el pensamiento humano. Se trata de que tome decisiones, aprenda y resuelva problemas como un ser humano. Dentro de este campo se encuentran las redes neuronales, por ejemplo.

- **Sistemas que actúan como humanos**

Estos pretenden imitar el comportamiento humano. Tratan de implementar comportamientos que, hasta ahora, nosotros los realizábamos mejor. Aquí encontramos a la robótica.

- **Sistemas que piensan racionalmente**

Estos tratan de imitar el pensamiento lógico racional que nos caracteriza a los seres humanos. Encontramos, por ejemplo, los sistemas expertos que son sistemas capaces de tomar decisiones como el ser humano.

- **Sistemas que actúan racionalmente**

Estos sistemas tratan de emular de forma racional el comportamiento humano. Aquí encontramos los agentes inteligentes que son entidades capaces de leer su entorno mediante sensores y tomar una decisión de forma racional, es decir, de forma correcta, actuando sobre él y maximizando la probabilidad de obtener el resultado deseado. (Torra 2011) (Wikipedia 2019)

2.16.2.2 Machine Learning

El Machine Learnig es la evolución de la inteligencia artificial que, además de poder responder siempre igual ante los mismos parámetros de entrada como la IA, es capaz de autoaprender y corregir errores.

2.16.2.2.1 ¿Cómo funciona?

Como su nombre indica, este sistema tiene que ser capaz de aprender y para ello se necesita la capacidad de generalizar y asociar. En el mundo de las máquinas esto se traduce en que tienen que realizar tareas con precisión y exactitud tanto en tareas conocidas como en nuevas o imprevistas.

Para ello se utilizan una serie de algoritmos que son capaces de aprender de los datos que se le están enseñando. Estos datos no se memorizan, sino que se utilizan para entrenar el algoritmo que se esté utilizando para prepararlo para situaciones que no estaban explícitamente en esos ejemplos.

Estos algoritmos se dividen en: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo.

2.16.2.2.2 Tipos de Machine Learning

- **Aprendizaje supervisado**

Utilizamos la información de un dataset de entrenamiento. Se trata de conseguir que diferencie entre objetos dentro de una imagen, por ejemplo. Para ellos se le enseñaran miles de imágenes etiquetadas, es decir que se dice en la imagen que es cada objeto, para que nuestro sistema lo aprenda. Después de este entrenamiento, nuestro sistema podrá determinar en imágenes que no le hemos enseñado si el objeto que aparece es una cosa u otra. A este problema se le llama clasificación. Otro ejemplo es predecir un valor continuo. A partir del código postal, el tamaño y el número de cuartos de una casa, se puede predecir cuál será su valor. A este problema se le conoce como regresión.

Lo que caracteriza este tipo es que se le entrena con muchos datos y a partir de ahí es capaz de generalizar para nuevos casos que se le presenten.

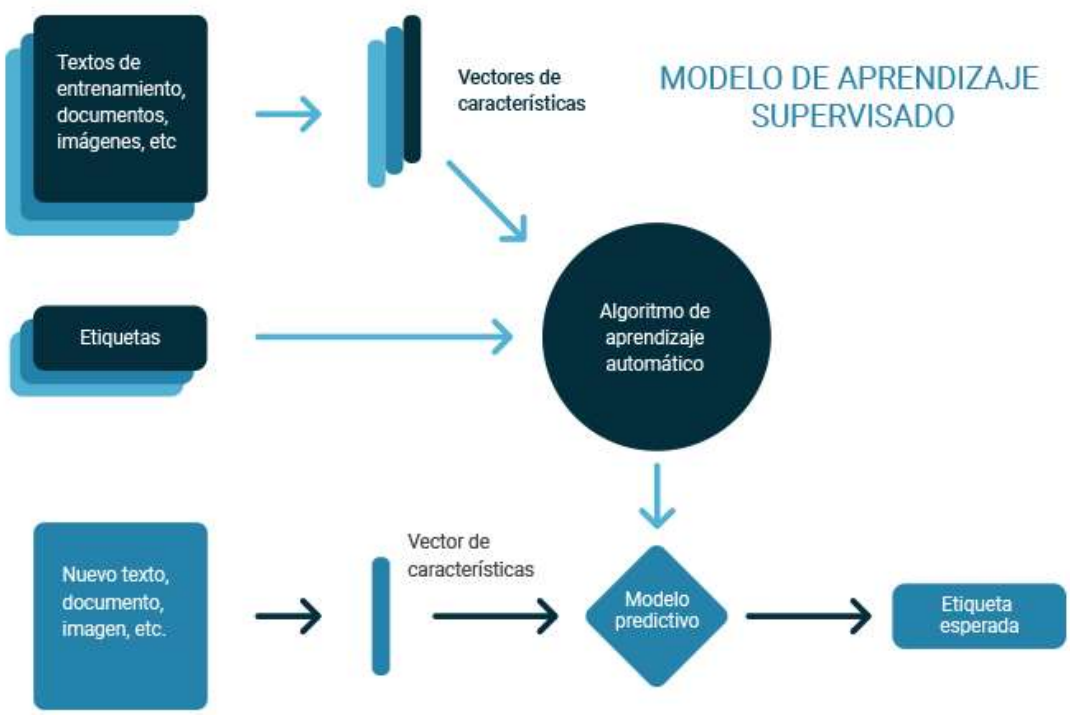


Figura 31: Esquema de funcionamiento del aprendizaje supervisado

- **Aprendizaje no supervisado**

A diferencia de la otra técnica, en esta no tenemos etiqueta. Estos modelos tienen el objetivo de comprender y abstraer los patrones de la información directamente. Por ejemplo, un modelo recibirá la ubicación de todas las fincas que posee un agricultor y determinará cómo separarlas en grupos para así lograr una mayor eficiencia. A este problema se lo conoce como “clustering”.

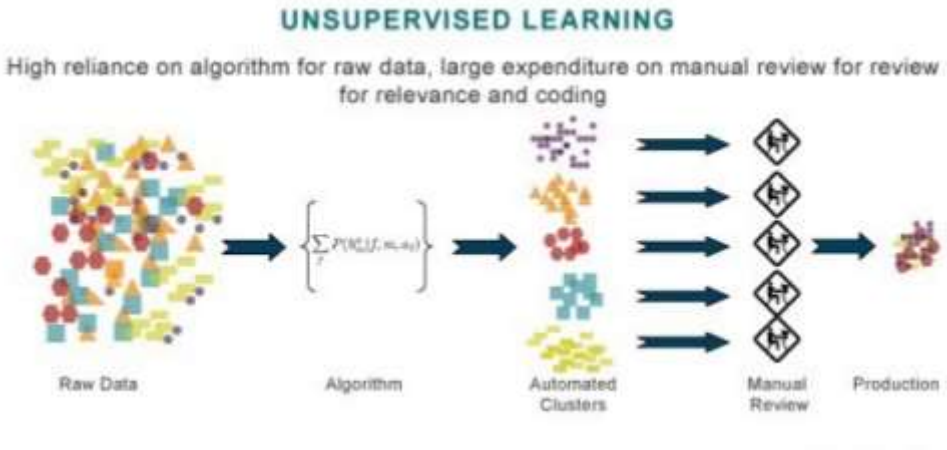


Figura 32: Esquema de funcionamiento del aprendizaje no supervisado

- **Aprendizaje por refuerzo**

Esta técnica se basa en la experiencia. Cada vez que el algoritmo lo hace bien se le da un “premio” y cada vez que lo hace mal se le “castiga”. A partir de estos premios y castigos se va optimizando y va aprendiendo cuál es la forma correcta de realizar sus tareas. La ventaja de esta técnica es que no requiere gran cantidad de datos.

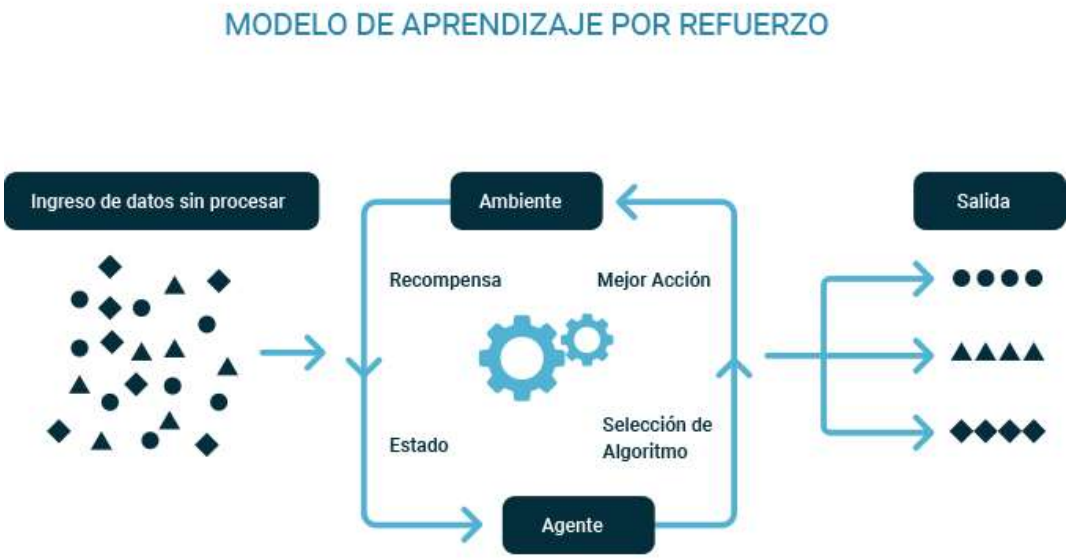


Figura 33: Esquema de funcionamiento del aprendizaje por refuerzo

Figura 34: Tabla resumen de los tres tipos de algoritmos de Machine Learning

(Kent 2019), (Sanseviero 2018)

2.16.2.3 Deep Learning

Es un subconjunto dentro del campo del Machine Learning, el cual predica con la idea del aprendizaje desde el ejemplo. La principal diferencia entre estos dos campos es que el Deep Learning leva el aprendizaje a un nivel más detallado.

En Deep Learning, en lugar de enseñarle a ordenador una lista enorme de reglas para solventar un problema, le damos un modelo que pueda evaluar ejemplos, así como una pequeña colección de instrucciones para modificar el modelo cuando se produzcan errores. Con el tiempo esperamos que esos modelos sean capaces de solucionar el problema de forma extremadamente precisa, gracias a que el sistema es capaz de extraer patrones.

Aunque existen distintas técnicas para implementar Deep Learning, una de las más comunes es simular un sistema de redes artificiales de neuronas dentro del software de análisis de datos.

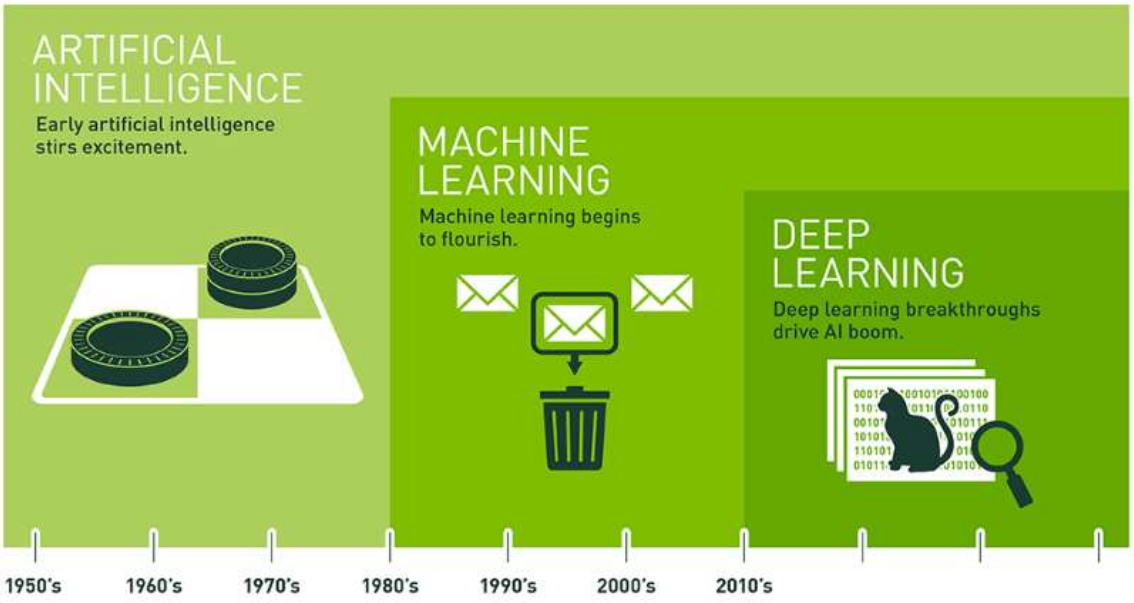


Figura 35: Tecnologías ordenadas cronológicamente

(Bejerano 2017), (Rodrígez 2018)

2.16.2.4 Redes Neuronales Artificiales (RNA)

Son un modelo computacional basado en el funcionamiento del cerebro humano. Consiste en un conjunto de neuronas artificiales que se interconectan entre sí para transmitirse la información. La información de entrada a esta red es sometida a varias operaciones produciendo uno o varios valores a su salida.

Cada neurona está conectada a otras a través de unos enlaces. Estos enlaces tienen el valor de salida de la neurona anterior, el cual es multiplicado por el peso que tenga esa conexión. Este peso puede aumentar, disminuir o inhibir el valor de resultado de la neurona anterior. Del mismo modo, a la salida de la neurona puede existir una funciona limitadora que modifica o impone un valor mínimo necesario para propagar la información a la siguiente neurona. A esta función se la conoce como función de activación.

Cada neurona hace una operación de regresión lineal, por lo que la red completa hace una concatenación de operaciones de regresión lineal. Matemáticamente se puede demostrar que la concatenación de todas estas líneas rectas da como resultado una línea recta por lo que sería como si solo hubiésemos hecho una única operación, es decir, que nuestra red neuronal colapsaría a una única neurona. Para evitar eso, debemos introducir en esa concatenación de rectas, alguna perturbación no lineal para evitar este colapso. Esta perturbación la añade la ya mencionada función de activación de cada neurona.

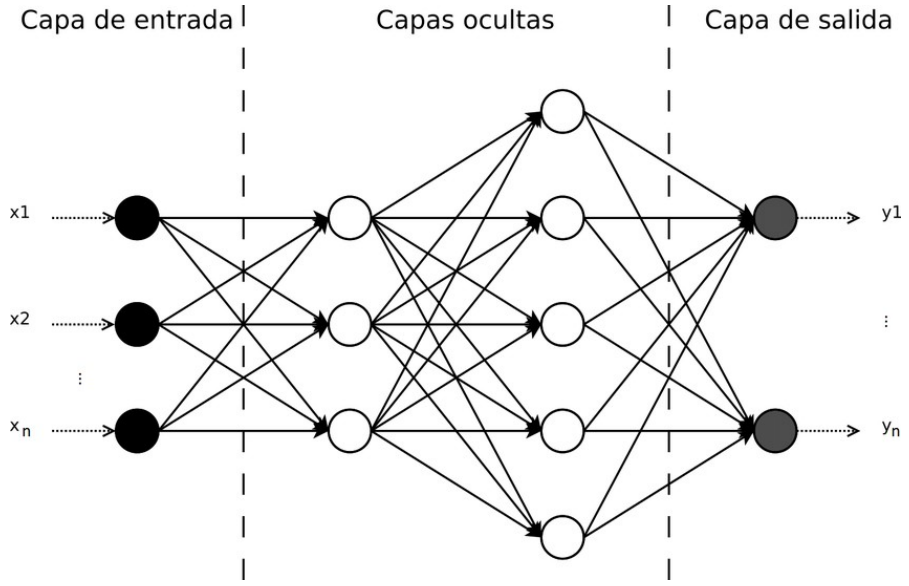


Figura 36: RNA con capa de entrada, dos capas ocultas y capa de salida

2.16.2.4.1 La neurona artificial

Las neuronas artificiales son la unidad básica de las RNA que intentan modelar el comportamiento de una neurona biológica. Hay varios modelos de neuronas artificiales pero la más común es la neurona de McCulloch-Pitts.

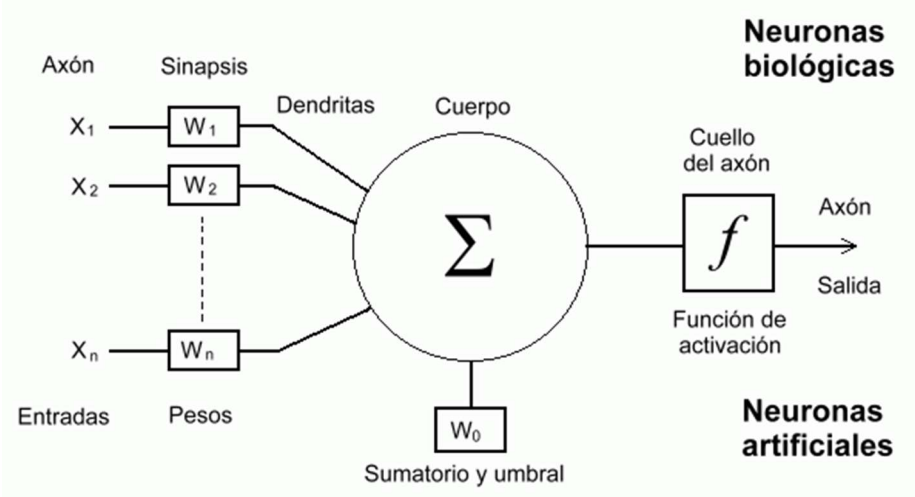


Figura 37: Neurona de McCulloch-Pitts haciendo comparación entre la neurona artificial y la neurona biológica

- Las entradas X_i reciben los datos de otras neuronas. En una neurona biológica corresponden a las dendritas.
- Los pesos sinápticos W_i. Al igual que en una neurona biológica se establecen sinapsis entre las dendritas de una neurona y el axón de otra, en una neurona artificial a las entradas que vienen de otras neuronas se

les asigna un peso. Este peso, que es un número, se modifica durante el entrenamiento de la red neuronal, y es aquí por tanto donde se almacena la información que hará que la red sirva para un propósito u otro.

- Regla de propagación. Es el cálculo que realiza la neurona.

$$Y_i = f\left(\sum_{i=1}^{n} W_i * X_i + W_0\right)$$

(Wikipedia 2019), (Justiniano 2018, 68)

2.16.2.4.2 Función de activación

Como se ha descrito antes, esta función introduce una perturbación no lineal necesaria para evitar el colapso de todas las neuronas de nuestra red. Esta función se elige de tal manera que sus derivadas sean sencillas de calcular para ahorrar tiempo de cálculo en los algoritmos de optimización, los cuales se explicaran más adelante. Las funciones de activación más comunes son:

→ **Sigmoide**

Esta función transforma los valores introducidos a una escala de entre 0 y 1 donde los valores bajos tienden a 0 y los altos a 1. Sus características son:

- Lenta convergencia
- No está centrada en 0
- Esta acotada entre 0 y 1
- Buen rendimiento en la última capa

Función:

$$f(x) = \frac{1}{1 - e^{-x}}$$

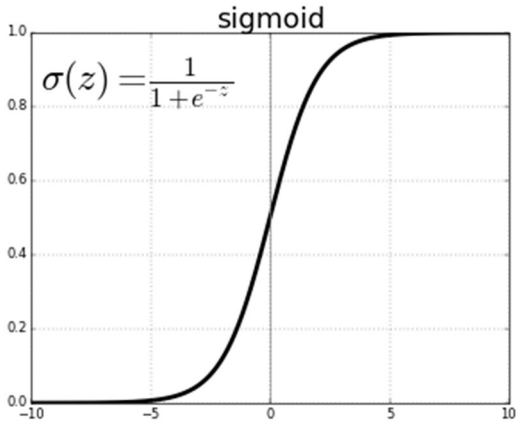


Figura 38: Función sigmoide

→ **Tanh – Tangente hiperbólica**

Esta función transforma los valores introducidos a una escala de entre -1 y 1 donde los valores bajos tienden a -1 y los altos a 1. Sus características son:

- Similar a la sigmoide
- Lenta convergencia
- Centrada en 0
- Está acotada entre -1 y 1
- Se utiliza para decidir entre una opción y la contraria
- Buen desempeño en redes recurrentes

Función:

$$f(x) = \frac{1}{1 + e^{-2x}} - 1$$

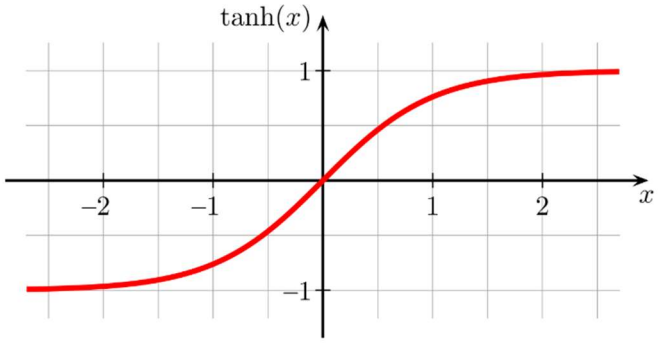


Figura 39: Función tangente hiperbólica

→ **ReLU**

Esta función anula los valores de entrada negativos y deja pasar los positivos sin modificarlos. Sus características son:

- No está acotada.
- Se pueden morir demasiadas neuronas.
- Se comporta bien con imágenes.
- Buen desempeño en redes convolucionales.

Función:

$$f(x) = max(0, x) = \ \{0 \ for \ x < 0 \ x \ for \ x \ \geq 0$$

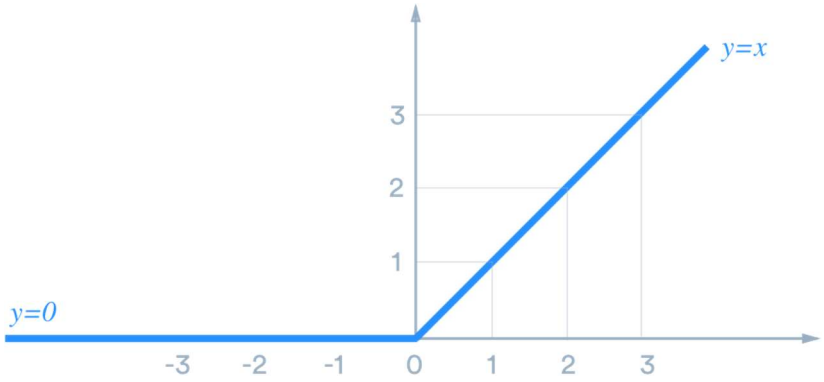


Figura 40: Función ReLU

→ **Leaky ReLU**

Esta función multiplica los valores de entrada negativos por un valor corrector y los valores positivos los deja pasar sin modificarlos. Sus características son:

- Similar a la función ReLU.
- Penaliza los negativos mediante un coeficiente rectificador.
- No está acotada.
- Se comporta bien con imágenes.
- Buen desempeño en redes convolucionales.

Función:

$$f(x) = \ \{a * x \ for \ x < 0 \ x \ for \ x \ \geq 0$$

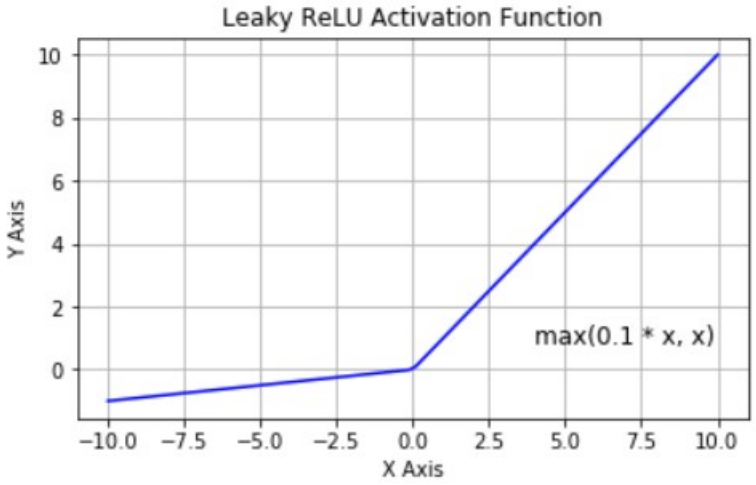


Figura 41: Función Leaky ReLU

→ **Softmax**

Esta función transforma las salidas a una representación en forma de probabilidades, de tal manera que el sumatorio de todas las probabilidades de las salidas da como resultado 1. Sus características son:

- Se utiliza cuando queremos tener una representación en forma de probabilidades.
- Es muy diferenciable.
- Buen rendimiento en las últimas capas.

Función:

$$f(Z) = \frac{e^{Zj}}{\sum_{k=1}^{K} e^{Zk}}$$

(Calvo 2018), (Justiniano 2018, 69 - 70)

2.16.2.4.3 Topología de red

Es la estructura de conexión de neuronas que puede adoptar nuestra red. Existen varios tipos, pero los podemos clasificar en tres.

→ **Feedforward**

Esta es la configuración de red más simple ya que cada neurona se conecta solo a la siguiente capa y así hasta que llegan a la última capa. Por lo que las neuronas de una misma capa no pueden conectarse entre ellas o conectarse a capas posteriores de su capa posterior.

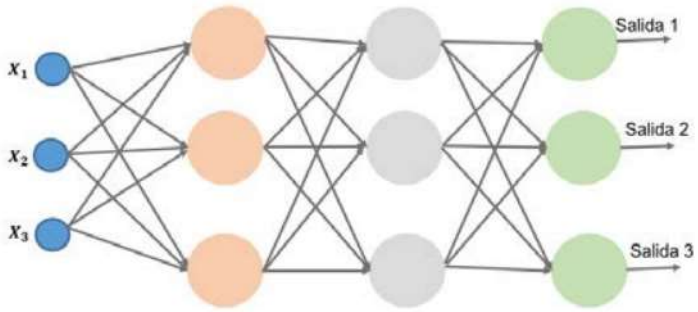


Figura 42: RNA con estructura feedforward

→ **Redes recurrentes**

Esta topología de red permite conexiones arbitrarias entre neuronas, incluso pudiendo crear ciclos. Con esto conseguimos temporalidad, permitiendo que la red tenga memoria.

Este tipo de redes son muy potentes para el análisis de texto, video o sonido.

- **Elman o recurrentes simples**

Estas redes incorporan retroalimentación con lo que se consigue que la red tenga memoria. Con esta topología conseguimos reducir el error utilizando un menor número de iteraciones gracias a que las neuronas conocen el valor anterior obtenido. En redes con muchas neuronas, el entrenamiento de estas se hace muy lento.

Son muy utilizadas para problemas de reconocimiento de voz o reconocimiento de escritura a mano.

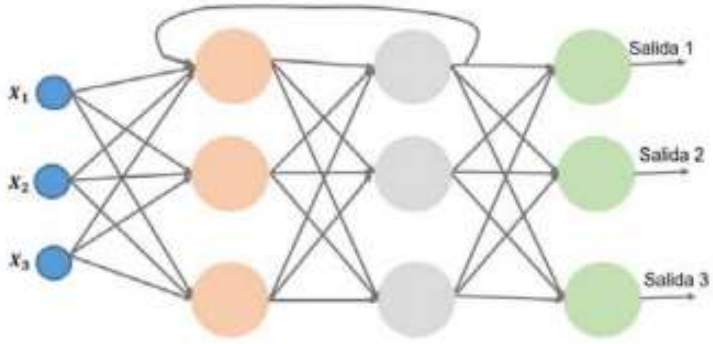


Figura 43: RNA con topología Elman

- **Recurrentes LSTM**

Las redes LSTM (Long Short Term Memory) están compuestas por unidades LSTM y son un tipo especial de red neuronal recurrente descritas en 1997 por Hochreiter & Schmidhuber.

El problema de las redes recurrentes convencionales es que presentan problemas en su entrenamiento por lo que al cabo del tiempo van acumulando error. Esta acumulación de errores provoca dificultades para memorizar dependencias a largo plazo.

Estos problemas son solucionados por estas redes LSTM las cuales deciden qué datos van a almacenar y cuales borrar.

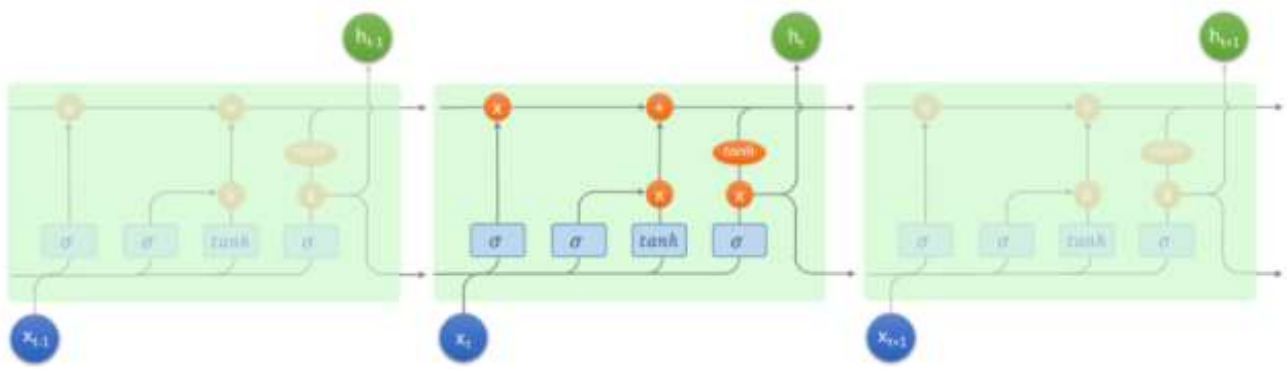


Figura 44: Red LSTM

Esta unidad de memoria contiene tres puertas que controlan cómo fluye la información:

- La puerta de entrada controla cuando puede entrar la nueva información.
- LA puerta del olvido controla cuando se olvida una parte de la información lo que le permite discriminar entre los datos importantes y los superfluos.
- La puerta de salida controla cuando se utiliza en el resultado de los recuerdos almacenados en la celda.

Este tipo de redes se utiliza en compresión de textos de lenguajes naturales, reconocimiento de voz, reconocimiento de gestos, reconocimiento de escritura a mano y captura de imágenes.

- **Redes GRU**

Las redes GRU (Gated recurrent unit) están compuestas por unidades GRU y son un tipo especial de red neuronal recurrente descritas en 2014 por Kyunghyun Cho et al.

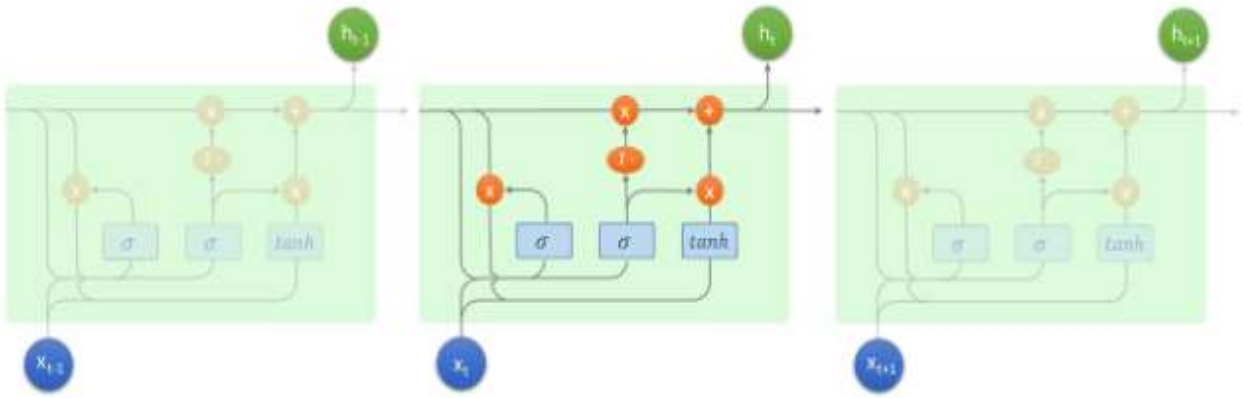


Figura 45: Red GRU

Esta unidad de memoria contiene dos puertas que controlan cómo fluye la información:

- La puerta de actualización indica cuanto del contenido de las celdas anteriores hay que mantener.
- La puerta de reajuste define cómo incorporar la nueva entrada con los contenidos anteriores de la celda.

Estas redes presentan un muy buen rendimiento en conjuntos de datos pequeños. Para grandes conjuntos de datos, se recomienda utilizar las redes LSTM. (Calvo 2018), (Justiniano 2018, 71 - 72)

2.16.2.4.4 Entrenamiento

Cuando creamos una nueva red neuronal los pesos de las conexiones entre neuronas se establecen de forma aleatoria por lo que nuestra red no será capaz de realizar su cometido correctamente. Si le introducimos valores a la capa de entrada obtendremos un resultado aleatorio en la salida, que puede ser correcto o no. Para hacer que nuestra red realice su cometido correctamente debemos ajustar esos pesos para obtener los resultados deseados y para ello debemos entrenar nuestra red. Para entrenar a nuestra red nos debemos ayudar de algún algoritmo que la optimice. Existen varios métodos de optimización como son el de perturbación aleatoria, descenso del gradiente, el método de Newton, el gradiente conjugado, el método cuasi-Newton o el algoritmo de Levenberg-Marquardt. Nos centraremos únicamente en el algoritmo de perturbación aleatoria, porque fue el primero que se utilizó, y el algoritmo del descenso del gradiente ya que es el más utilizado.

- **Método de perturbación aleatoria**

Es un método de fuerza bruta que consistía en ver cuanto variaba nuestra función de costes, nuestra salida, cuando hacíamos pequeñas variaciones en los parámetros de nuestra red. Este método era eficiente cuando teníamos muy pocas neuronas ya que tenemos pocos caminos que recorrer para llegar a la última capa. Pero, cuantas más neuronas tiene nuestra red, se añaden más parámetros lo que crea más caminos que nos conducen hasta la última capa y más iteraciones tenemos que hacer para ver como varia nuestra función de coste.

Debido a este problema, se hace inviable optimizar una red neuronal con miles de neuronas mediante este método ya que la capacidad computacional necesaria para llevar a cabo este algoritmo de optimización crece exponencialmente cuantas más capas añadimos a nuestra red.

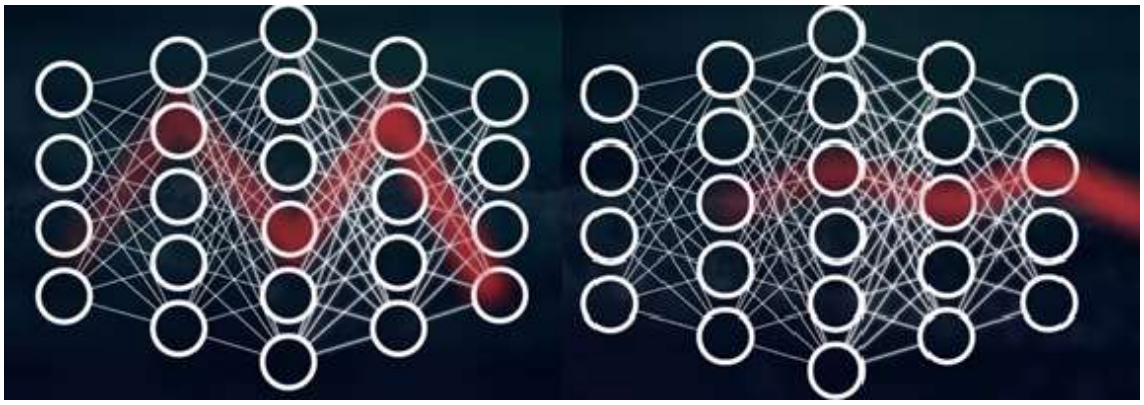


Figura 46: Caminos utilizados por un algoritmo de perturbación aleatoria

- **Método del descenso del gradiente**

El vector gradiente de cualquier función nos indica hacia donde crece más rápido. Esto se obtiene calculando las derivadas parciales de cada parámetro de nuestra función. La función de costes de nuestra red neural puede tener desde uno a miles de parámetros.

Lo que a nosotros nos interesa es que el error de la función de costes sea mínimo. Para ello utilizamos el concepto que hemos descrito antes, el vector gradiente. Como el gradiente nos indica hacia donde nuestros parámetros, crecen más rápidamente, deberemos restarles este resultado para así ir descendiendo en nuestra función hasta alcanzar un mínimo relativo, es decir, reducir el error al mínimo posible.

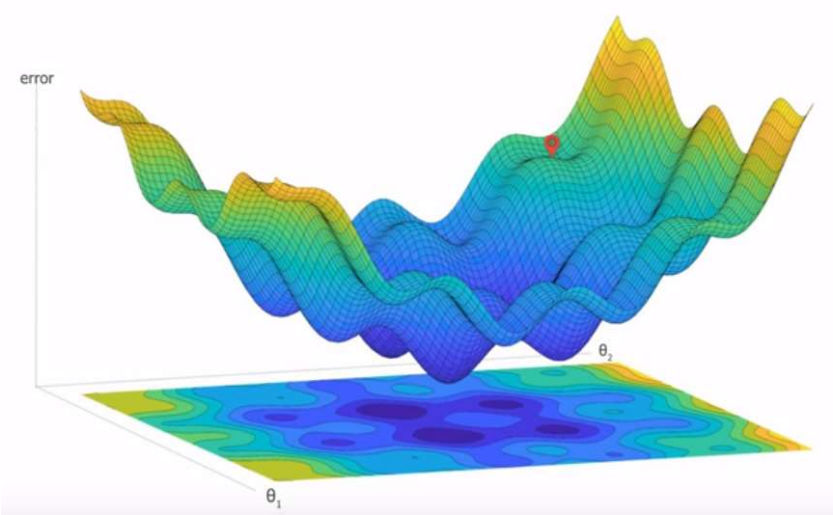


Figura 47: Función de costes de una red neuronal con dos parámetros

En esa imagen podemos observar la función de costes de una red neuronal con dos parámetros y el punto en el que nos encontramos, señalado con una marca roja. Este punto es aleatorio ya que cuando creamos una RNA los valores iniciales son aleatorios. Desde ese punto debemos ir bajando en nuestra función, ayudándonos del vector gradiente, hasta que encontremos un mínimo relativo. Observando la imagen vemos que, si alcanzamos el mínimo absoluto de esa función, tendremos un error pequeño, como nos lo indica el eje "error". Matemáticamente quedaría de la siguiente manera:

- Primero calculamos el vector gradiente (∇f)

$$\nabla f = \begin{pmatrix} \frac{\partial error}{\theta_1} \\ \frac{\partial error}{\theta_2} \\ \vdots \\ \frac{\partial error}{\theta_n} \end{pmatrix}$$

- Restamos el gradiente a nuestros parámetros

$$\theta = \theta - \nabla f$$

- Por último, repetir los dos puntos anteriores iterativamente hasta que lleguemos a un punto donde movernos ya no suponga una variación notable del coste, es decir, que la

pendiente es casi nula o lo que es lo mismo, habremos alcanzado un mínimo local.

Ahora bien, para tener completo nuestro algoritmo, debemos añadir un parámetro más que se define como ratio de aprendizaje (α). Este parámetro nos indica cuanto avanzamos en cada iteración. La elección correcta de este parámetro es fundamental porque si los pasos son muy grandes en cada iteración, puede que nuestro algoritmo nunca converja y, en cambio, si elegimos un valor muy pequeño, tardaremos demasiado tiempo en optimizar nuestra función. Con este parámetro, el cálculo que se debe hacer es el siguiente:

$$\theta = \theta - \alpha \nabla f$$

A modo de ejemplo, se van a exponer tres imágenes que muestran el intento de alcanzar un mínimo en una función con dos parámetros mediante este algoritmo. La línea roja indica por donde ha ido pasando nuestro punto y las zonas del mapa más oscuras indicarían los mínimos de la función.

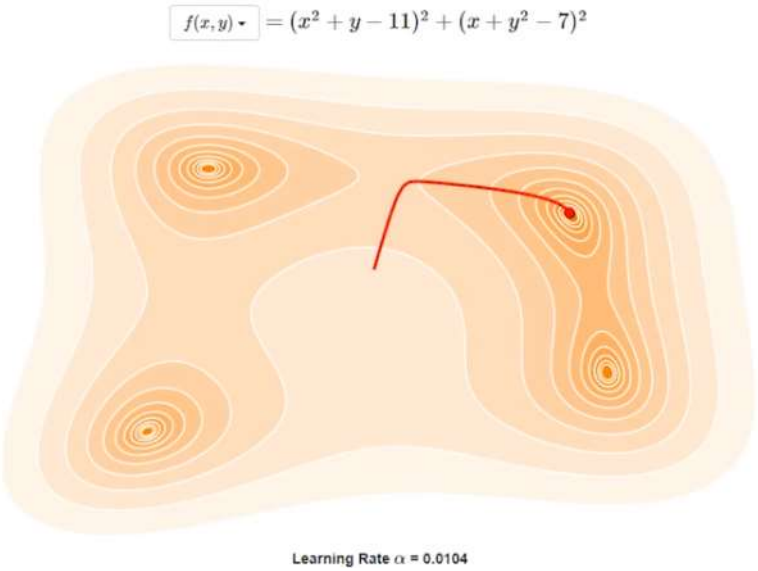


Figura 48: Función optimizada correctamente

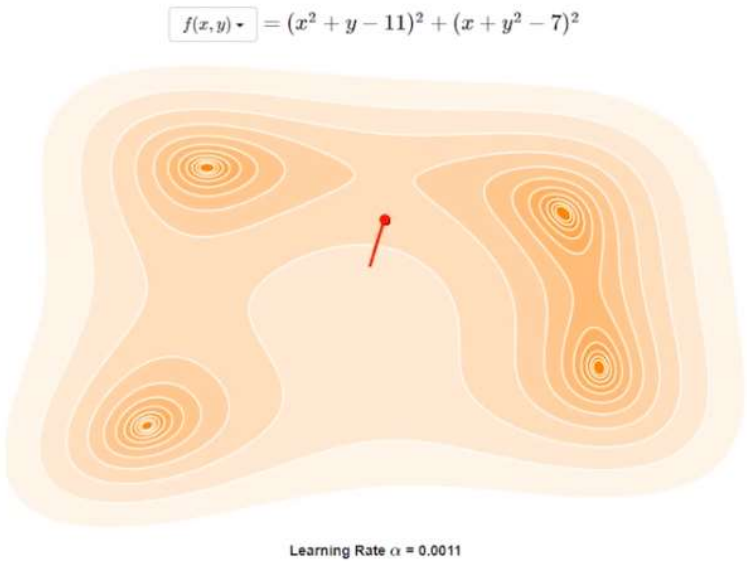


Figura 49: Función que no consigue ser optimizada en un tiempo razonable porque el ratio de aprendizaje es muy pequeño

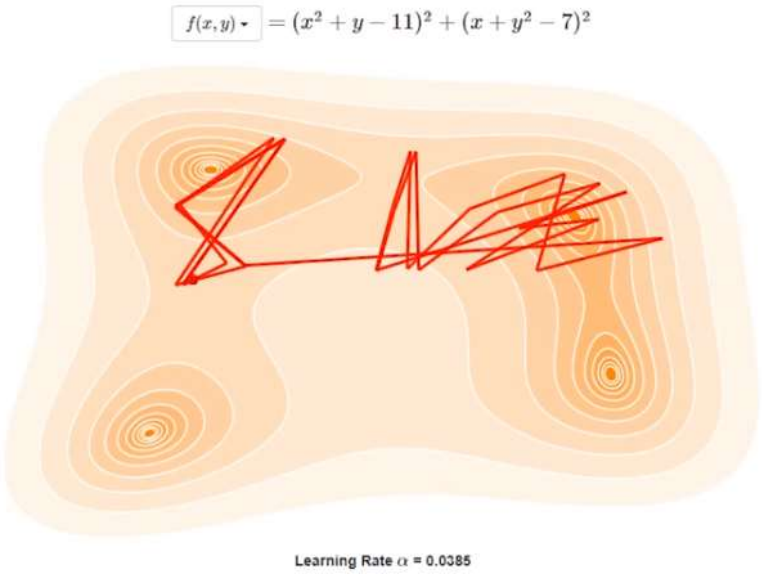


Figura 50: Función que no consigue ser optimizada porque nunca terminaran las iteraciones debido a que el ratio de aprendizaje es demasiado elevado

(CSV 2018)

2.16.2.4.5 Backpropagation

Calcular las derivadas parciales de los parámetros de nuestra red neuronal es sencillo cuando hay pocos. Por ejemplo, calcularlas de una única neurona con 3 parámetros de entrada es sencillo, ya que la función de coste solo depende de estos y las derivadas parciales $\left(\frac{\partial f.costes}{\theta}\right)$ son sencillas de calcular.

Pero cuando tenemos muchas neuronas y capas, la derivada parcial de la función de costes respecto a los parámetros de nuestra red ya no es tan sencilla de calcular porque la implicación que tiene cada uno de los parámetros en el resultado depende de los parámetros que están antes y después de él. Para obtener este cálculo utilizamos el algoritmo de backpropagation.

El fundamento de este algoritmo es ir propagando el error, resultado de una iteración de nuestra red, hacia atrás en las capas de nuestra red para conseguir dar con la o las neuronas que más responsabilidad han tenido en ese resultado y así modificar sus parámetros para que ir eliminando ese error.

Para la explicación matemática volveremos a nuestra neurona, la cual tiene dos parámetros de entrada y el parámetro de vallas.

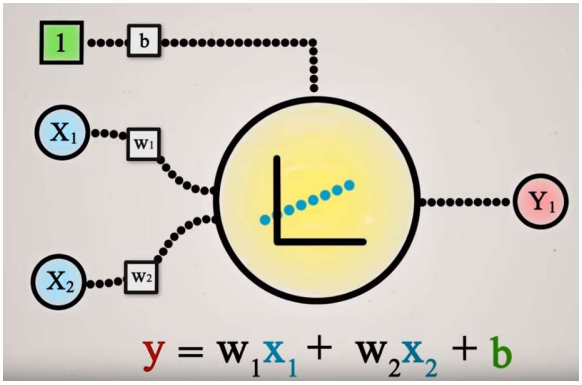


Figura 51: Neurona simple o perceptrón con dos parámetros de entrada más el de vallas

Para esta neurona deberemos calcular la derivada parcial del coste respecto de los pesos y la derivada parcial del coste respecto al parámetro de vallas.

$$\frac{\partial C}{\partial w} \quad \frac{\partial C}{\partial b}$$

Ahora calcularemos el valor de estas derivadas para las neuronas de nuestra última capa. Marcaremos con un superíndice “L” para indicar que esta es nuestra última capa suponiendo que nuestra red tiene L capas. Para llevar a cabo este cálculo deberemos analizar que conecta a “C” con nuestros parámetros.

El resultado de cada neurona es una suma ponderada de sus parámetros de entrada a la cual la llamaremos “Z”. Esta suma pasa a través de nuestra función de activación “a” y esta última es evaluada por la función de coste “C” la

cual nos da el error. Como podemos observar esto es una composición de funciones. Si la particularizamos para la última capa, nos queda:

$$C\left(a^L(Z^L)\right)$$

$$Z^L = W^L X + b^L$$

Para calcular la derivada de una composición de funciones se utiliza la regla de la cadena. Entonces nos quedaría como:

$$Z^L = W^L a^{L-1} + b^L$$

$$\frac{\partial C}{\partial w^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial w^L}$$

$$\frac{\partial C}{\partial b^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial b^L}$$

Ahora debemos calcular cada una de esas derivadas parciales. La primera derivada $\left(\frac{\partial C}{\partial a^L}\right)$ nos está pidiendo calcular la derivada de la función de coste respecto a la salida de nuestra red (a^L). Si por ejemplo nuestra función de coste es el error cuadrático medio:

$$C\left(a_j^L\right) = \frac{1}{2}\sum_{j=1}^{n}\left(y_j - a_j^L\right)^2$$

La derivada seria:

$$\frac{\partial C}{\partial a^L} = \left(a_j^L - y_j\right)$$

La siguiente derivada, la derivada de la activación con respecto a "z" $\left(\frac{\partial a^L}{\partial z^L}\right)$, nos esta indicando cuanto varia la salida de la neurona cuando variamos la suma ponderada de la neurona. Esto seria calcular la derivada de la función de activación. Suponiendo que es una sigmoide, obtendremos:

$$a^L(z^L) = \frac{1}{1 + e^{-z^L}}$$

$$\frac{\partial a^L}{\partial z^L} = a^L\left(z^L\right) \cdot \left(1 - a^L\left(z^L\right)\right)$$

Y por último tendríamos las derivadas que nos indican como varia "z" en función de los parámetros de entrada $\left(\frac{\partial z^L}{\partial w^L}, \frac{\partial z^L}{\partial b^L}\right)$. Sus resultados son:

$$z^L = \sum_{i=1}^{n}\left(a_i^{L-1} w_i^L + b^L\right)$$

$$\frac{\partial z^L}{\partial w^L} = a_i^{L-1}$$

$$\frac{\partial z^L}{\partial b^L} = 1$$

Ya solo nos quedaría juntar todas estas derivadas y obtendríamos la derivada deseada. Pero antes se va a explicar el concepto del error imputado.

Si observamos las ecuaciones descritas podemos ver que:

$$\frac{\partial C}{\partial z^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L}$$

Esta derivada nos está indicando cuanto varía el coste, el error, cuando se produce una variación en los parámetros de la neurona. Por lo que esto nos indica que sí la derivada da como resultado un número grande, significa que un cambio pequeño en la entrada se verá muy reflejado en el resultado final y, por el contrario, si el resultado es pequeño, ante una variación pequeña de los parámetros no veremos un gran cambio en la salida. Resumiendo, esta derivada nos está indicando qué responsabilidad tiene la neurona en el resultado final. A esta derivada se la conoce como el error imputado a la neurona y se representa "δ^L".

Si ahora reescribimos las derivadas anteriores, nos quedarían como:

$$\frac{\partial C}{\partial w^L} = \delta^L a_i^{L-1}$$

$$\frac{\partial C}{\partial b^L} = \delta^L$$

$$\delta^L = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L}$$

Este cálculo nos sirve para calcular las derivadas parciales de la última capa, pero estas dependen a su vez de la capa anterior y estas últimas de la anterior y así sucesivamente por lo que debemos calcular todas estas para obtener el resultado deseado. Para ello empezamos haciendo lo mismo que hasta ahora, pero con la capa L – 1:

$$C\left(a^L(W^L a^{L-1}(W^{L-1}a^{L-2} + b^{L-1}) + b^L)\right)$$

$$\frac{\partial C}{\partial w^{L-1}} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial a^{L-1}} \cdot \frac{\partial a^{L-1}}{\partial z^{L-1}} \cdot \frac{\partial z^{L-1}}{\partial w^{L-1}}$$

$$\frac{\partial C}{\partial b^{L-1}} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial a^{L-1}} \cdot \frac{\partial a^{L-1}}{\partial z^{L-1}} \cdot \frac{\partial z^{L-1}}{\partial b^{L-1}}$$

Observando las ecuaciones vemos que el primer y el segundo miembro se corresponde con el error imputado, el tercer miembro es la matriz de costes “W” que conecta ambas capas, el cuarto miembro es la derivada de la función de activación de la capa anterior y el último miembro es el mismo resultado de antes, 1 y a_i^{L-2} respectivamente. De esta manera, a los miembros uno, dos, tres y cuatro los nombraríamos el error imputado al igual que antes:

$$\frac{\partial C}{\partial z^{L-1}} = \delta^{L-1}$$

Y escribiríamos las ecuaciones como:

$$\frac{\partial C}{\partial w^{L-1}} = \delta^{L-1} a_i^{L-2}$$

$$\frac{\partial C}{\partial b^{L-1}} = \delta^{L-1}$$

Como vemos aquí, lo que estamos realizando es propagar el error hacia atrás. Con esta misma metodología podemos calcular las derivadas de las capas anteriores. Por lo que la metodología para aplicar este algoritmo es la siguiente:

1º. Computo del error de la última capa

$$\delta^l = \frac{\partial C}{\partial a^l} \cdot \frac{\partial a^l}{\partial z^l}$$

2º. Retropropagamos el error a la capa anterior

$$\delta^{l-1} = W^l \delta^l \cdot \frac{\partial a^{l-1}}{\partial z^{l-1}}$$

3º. Calculamos las derivadas de la capa usando el error

$$\frac{\partial C}{\partial w^{l-1}} = \delta^{l-1} a_i^{l-2}$$

$$\frac{\partial C}{\partial b^{l-1}} = \delta^{l-1}$$

4º. Repetimos el proceso, pero desde la capa anterior

Repetiríamos este proceso hasta el principio de nuestra red y así calcularíamos las derivadas deseadas para poder aplicar el algoritmo del descenso del gradiente. (CSV 2018), (CSV 2018)

2.16.2.4.6 Errores de entrenamiento

Al entrenar RNA puede darse el caso de que se sobreentrenen, es decir, nuestro modelo solo va a ser bueno resolviendo casos que ya se le han enseñado, fallando en cualquier otro caso. Esto se debe a que el modelo de red es muy complejo en relación con los datos de entrada, por lo que no tiene suficientes datos para capturar el modelo de datos subyacentes y pierde su capacidad de generalizar. A continuación, se muestra un problema de clasificación en el cual la red ha sido sobre entrenada:

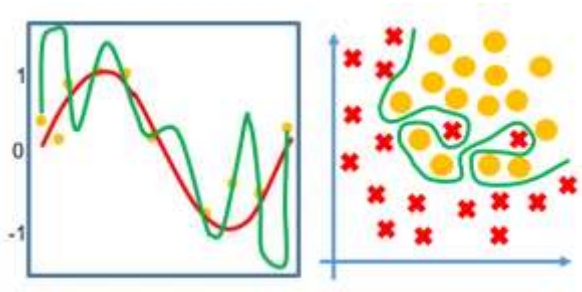


Figura 52: RNA sobreentrenada

Otro error que se puede obtener en el entrenamiento es que sean pocas las diferencias que existan entre los datos de entrada, generando así un modelo simple. Por lo que no hace una buena diferenciación entre ellas y ante un nuevo ejemplo, no hará bien esta clasificación. En esta imagen se puede ver este problema:

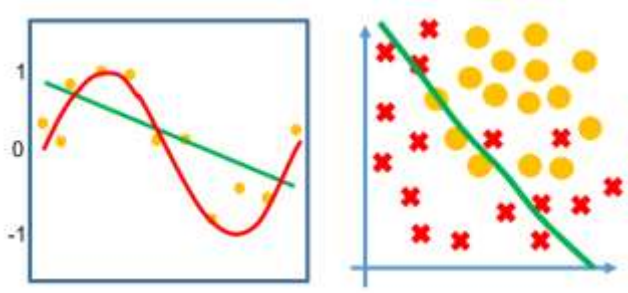


Figura 53: RNA con un entrenamiento poco efectivo

Por último, una RNA correctamente entrenada:

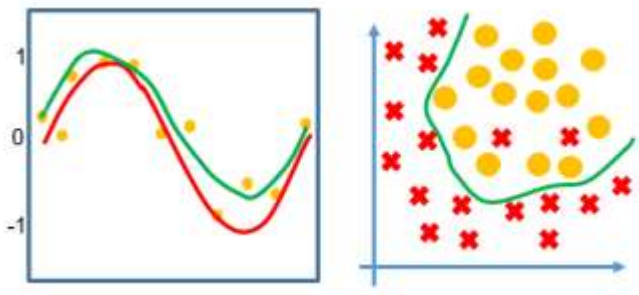


Figura 54: RNA correctamente entrenada

(Justiniano 2018, 72), (Caparrini 2018)

2.16.3 RNA en nuestro sistema

Teniendo ya los datos en nuestro microprocesador, deberemos prepararlos para su procesado y dar un resultado sobre lo que está ocurriendo actualmente. Esto se llevará a cabo a través de una red neuronal artificial. Esta red será programada utilizando cualquier herramienta que nos permita crear, analizar y entrenar una RNA.

Lo primero que se debe hacer es preparar la información con la que se va a alimentar la red ya que enviar a la red valores instantáneos no es útil porque lo que realmente nos indica que algo raro está ocurriendo es que se haya producido un cambio brusco en los sensores.

Para ello, nuestro microprocesador deberá realizar un cálculo antes de presentarle los datos a la RNA. El procedimiento será el siguiente:

- Realizará la media de las ultimas muestras tomadas de los sensores.
- Guardará esta media.
- Realizará la media de las siguientes muestras que se tomen.
- Restará ambas medias y hará el valor absoluto del resultado.

Realizando este tratamiento de los datos, nos aseguramos de que tendremos variaciones claras en los datos para que la red sea capaz de tomar decisiones en base a cambios que se producen en los sensores.

El número de muestras de las cuales se hará la media dependerá del tiempo de muestreo que se configure. Si éste es pequeño, la media se deberá realizar de más muestras y si es grande, la media se realizará de menos muestras.

Para el diseño de la RNA se utilizará una topología de MLP (perceptrón multicapa) porque este tipo de topología además de utilizar el aprendizaje supervisado para su entrenamiento da buenos resultados para encontrar la relación que existe entre unas entradas y unas salidas que es lo que necesitamos.

Esta red contará con una capa de entrada, la cual poseerá tantas neuronas como sensores queramos medir. En nuestro caso necesitaremos una neurona para el sensor de guiñada, dos parará cada uno de los sensores de inclinación, otras dos para los dos acelerómetros, otra para el de colisión, otra para el sensor de presión en el asiento y otra para los airbags. También harán falta neuronas para los sensores que se han comentado después de estos, como son el de llama, el o los de deformación, el de freno y el de giro de volante.

Para los primeros sensores que hemos comentado serían necesarias ocho neuronas. Para los segundos, serán necesarias tantas neuronas como sensores se instalen en el vehículo. Con estos segundos se deberán hacer

pruebas cuando se conecten a la red neuronal para ver su implicación en el resultado. Si su implicación es muy poca, se podrá considerar su eliminación. Por lo que la capa de entrada contará con las neuronas necesarias para los primeros sensores mas las necesarias para los segundos.

En la capa de salida se deberán colocar tantas neuronas como clasificaciones distintas queramos que se realicen. En nuestro caso serían necesarias cuatro. La primera sería para el funcionamiento normal, es decir, no hay accidente, la segunda para una colisión leve, la tercera para una colisión moderada y la última para un accidente grave. Además, las neuronas de esta capa final llevaran como función de activación la función sigmoidea que nos dará como resultado valores entre cero y uno. Esta función es más útil que, por ejemplo, una que nos daría un resultado binario, ya que podremos obtener mejores resultados sabiendo cuál de las salidas tiene un mayor porcentaje.

Las capas ocultas las crearemos por tanteo, es decir, empezaremos con pocas capas y pocas neuronas e iremos aumentando hasta obtener unos resultados coherentes y acertados. Comenzaremos con una sola capa y un número pequeño de neuronas, por ejemplo, cinco, y entrenaremos la red para ver el tiempo que tarda en entrenarse, cuánto error obtenemos, cuánto de acertados son los resultados etc. Iremos aumentando el número de capas y neuronas de forma progresiva hasta obtener respuestas muy acertadas y que el error sea pequeño. En este diseño deberemos tener en cuenta lo siguiente:

- El entrenamiento se hace más lento cuantas más capas ocultas y más neuronas se usen.
- Cada capa adicional hace que el gradiente sea más inestable.
- El número de mínimos locales de la función normalmente aumenta cuantas más capas ocultas utilicemos.
- Si utilizamos muchas neuronas en una sola capa oculta y no se resuelve el problema, significa que debemos utilizar más capas ocultas.
- Si se utiliza un número elevado de capas y/o neuronas existe mayor posibilidad de sobreentrenar la red, es decir, que sea muy buena resolviendo los casos que se le han enseñado, pero es muy mala para casos nuevos.
- Para redes con mayor número de entradas que de salidas, como la nuestra, se recomienda ir disminuyendo el número de neuronas de capa oculta en capa oculta hasta llegar a la capa final.

2.17 Bloque alimentación

Este sistema necesita de una fuente de alimentación externa para funcionar. La mayoría de los aparatos electrónicos que vamos a utilizar necesitan ser alimentados a 5v excepto dos módulos que serán alimentados a 12v.

Todos los vehículos tienen en su interior una batería, como mínimo, que nos da una tensión de 12v. Esta batería será la que utilicemos para alimentar nuestros dispositivos. Como la mayoría de aparatos necesitan 5v, será necesario convertir esa tensión a una tensión menor, 5v.

En el mercado ya existen aparatos que nos transforman la tensión de 12v a 5v por lo que no será necesario que la diseñemos. Un ejemplo sería el siguiente, que viene preparado para montarlo en un coche.

Figura 55: Convertidor 12v a 5v 3A

También, un añadido que sería importante añadir es una batería extra con 12v de salida también que se cargue con la batería que lleva el propio coche. Sería interesante puesto que en caso de accidente es muy posible que la batería del vehículo se dañe y deje de funcionar por lo que nos quedaríamos sin alimentación en nuestro sistema por lo que sería inútil. Teniendo una batería extra que se encuentre en el compartimento donde está todo nuestro sistema, el cual estará protegido, nos brindará la energía necesaria para que siga funcionando para que se realicen todas las funciones de emergencia.

2.18 Bloque de almacenaje de datos

Un elemento muy importante que necesita este sistema es una memoria no volátil para guardar el informe cuando se produce la colisión. Para ello,

utilizaremos una memora microSD y un módulo compatible con Arduino que nos permite leer y escribir sobre ella, en nuestro caso solo será necesario escribir.

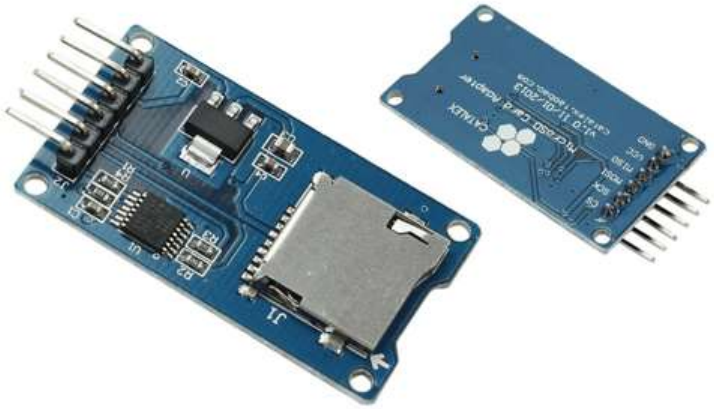


Figura 56: Modulo para Arduino con capacidad de leer y escribir en una tarjeta de memoria microSD

Este módulo deberá ser conectado a los pines SPI que incorpora nuestra placa. Es necesaria este tipo de conexión ya que el módulo la requiere. Primero deberemos colocar al ChipKIT como maestro. Esto se realiza poniendo los jumpers presentes en la placa como indica la imagen:

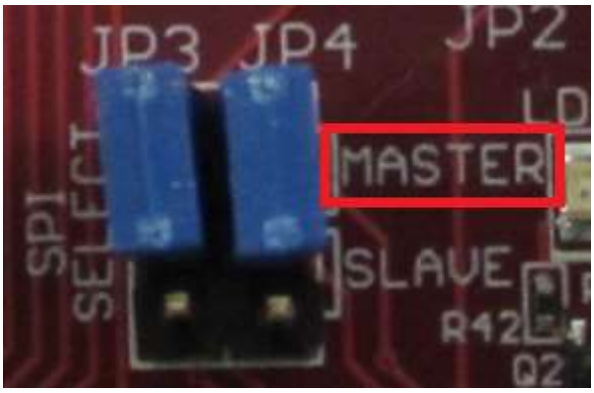


Figura 57: Colocación de los jumpers para que la placa trabaje como maestro

Después debemos conectar los pines como sigue:

- Los 5v del módulo a alimentación de 5v
- El GND del módulo al GND de la fuente
- El pin MISO al pin 50 de ChipKIT
- El pin MOSI al pin 51 de ChipKIT
- El pin SCK al pin 52 de ChipKIT
- El pin CS al pin 53 de ChipKIT

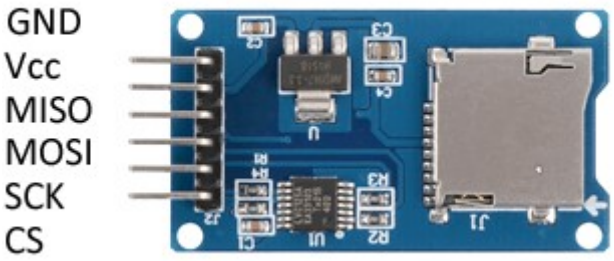


Figura 58: Designación de pines del módulo de microSD

Para utilizar este módulo de forma cómoda y fácil, podemos incluir a nuestro proyecto de Arduino su librería. Esta librería ya incorpora funciones para el manejo de ficheros y directorios.

Cuando el resultado de la RNA sea de nivel dos o más, accidente leve, moderado o grave, todos los datos de sensores guardados en la memoria volátil del microprocesador se deberán ser enviados a la tarjeta microSD. Se creará un archivo con fecha y hora de cuándo se va a realizar la escritura y dentro se meterán todos los datos.

2.19 Bloque de comunicación por radiofrecuencia

Una parte importante de este sistema es la de avisar a los coches circundantes para evitar otra posible colisión contra el vehículo ya accidentado. Para realizar esto existen varias formas y varios tipos de comunicaciones, pero la que mejor va a funcionar para nuestro cometido es la comunicación por radiofrecuencia. Este tipo de comunicación nos va a permitir avisar a los vehículos a más distancia de forma inalámbrica.

Para nuestro dispositivo utilizaremos los módulos RF433MHz emisor y receptor de radiofrecuencia.

Los módulos de radio frecuencia RF433MHz son transmisores/receptores inalámbricos que podemos emplear como forma de comunicación entre procesadores.

Operan a 433MHz, aunque también existen módulos similares a 315MHz. Ambas frecuencias pertenecen a bandas libres, por lo que su uso es gratuito.

El alcance depende del voltaje con el que alimentemos el módulo y la antena que usemos. A 5V y con la antena del módulo, el alcance difícilmente excederá de los 2 metros. Alimentando a 12V y con una antena de cobre de 16.5cm el rango en exteriores puede alcanzar 300 metros.

La comunicación es simplex y tienen baja velocidad de transmisión de alrededor de 2400bps. Se realiza por modulación ASK (amplitude shift keying). Como no disponen ni de filtro ni de ID, para establecer una comunicación robusta, deberemos implementarlos vía software. (Llamas 2019)

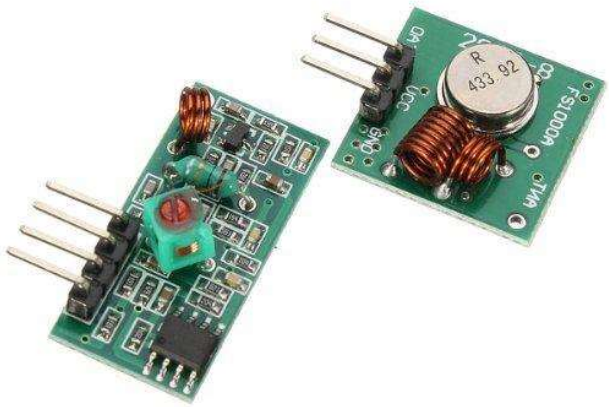


Figura 59: Módulo de transmisión y módulo de recepción RF 433MHz

Este sistema deberá llevar los dos ya que debe ser capaz de enviar datos por radiofrecuencia y recibirlos en caso de que otro vehículo este enviando datos.

Estos serán alimentados a 12 voltios y se les instalarán antenas de cobre para ampliar el rango lo máximo posible.

Para el desarrollo software de este módulo podremos utilizar la librería "VirtualWire.h" la cual dispone de las funciones necesarias para realizar la comunicación.

2.20 Bloque de comunicación con el conductor

Cuando la RNA de un resultado de 3, es decir, accidente moderado es necesario comunicarnos con el conductor para saber su estado y actuar de acuerdo con ello.

En este apartado tendremos dos tipos de comunicación, una directa, es decir, nos comunicaremos directamente con el conductor mediante un altavoz y él se comunicará con el sistema mediante botones, y otra indirecta que será para medir su pulso mediante una pulsera que él lleve en su muñeca.

Para comunicarnos directamente con el conductor utilizaremos un altavoz pequeño que se instalara dentro del habitáculo. Este dispositivo no es necesario

que sea muy complejo ya que sólo reproducirá un audio que es el que indicara al usuario que, si está en buenas condiciones, deberá desactivar el sistema de emergencia y si no contesta, se deberá activar el sistema automáticamente.

Para esto, utilizaremos el módulo ISD1820 con altavoz. Este módulo compatible con Arduino nos permite grabar un audio y luego reproducirlo por el altavoz.

Figura 60: Módulo ISD1820 más altavoz

Grabaremos un audio con este dispositivo y lo reproduciremos cuando sea necesario. Sera un audio simple y claro para avisar al conductor de que se va a desencadenar el sistema de emergencia.

Para que el conductor se comunique con nosotros será necesario instalar unos botones en el salpicadero o volante del vehículo para que pueda actuar sobre ellos. Dispondrá de dos botones, uno que es el propio SOS y otro que será para cancelar la actuación del sistema de seguridad en caso de pulsar el botón SOS por accidente o de que el sistema se haya comunicado con él porque se va a desencadenar el protocolo de emergencia.

El botón SOS tendrá dos funcionalidades, la primera será desencadenar el protocolo de emergencia. Este botón manual es útil porque no necesariamente debe haber ocurrido un accidente para que el conductor o sus ocupantes tengan una emergencia. Al pulsar este botón y pasados unos segundos para poder cancelar el protocolo con el otro botón, se avisará a los servicios de emergencia al igual que se hace de forma automática, pero sin comunicarse con el conductor puesto que este está forzando el protocolo y significa que lo necesita realmente.

La segunda funcionalidad será la de aceptar el protocolo de emergencia cuando el sistema se haya comunicado con nosotros. Si el sistema se ha

comunicado con nosotros significara que ha detectado colisión por lo que podremos pulsar este botón para

aceptarla. También se aceptará automáticamente en el caso de que el conductor no conteste en un tiempo determinado.

Para la comunicación indirecta, como se ha comentado, se utilizará una pulsera que sea capaz de leer el pulso de una persona y enviar los datos por bluetooth. Existen en el mercado infinidad de dispositivos que realizan esto, aparte de otras funciones, que podríamos utilizar. Todos estos dispositivos son de empresas privadas por lo que será necesario trabajar conjuntamente para comunicar nuestro sistema con estos dispositivos mediante bluetooth. Una pulsera que podemos utilizar es la Xiaomi Mi Band. Esta pulsera, aparte de otras funcionalidades que tiene relacionadas con nuestro teléfono móvil, es capaz de medirnos el pulso. Estos datos de ritmo cardíaco serán enviados a nuestro sistema vía bluetooth para ser utilizados cuando se vaya a enviar un informe a los servicios de emergencia.

Como nuestro ChipKIT no dispone de conexión bluetooth deberemos añadirle un módulo de comunicaciones bluetooth. Para ello añadiremos el módulo DSD TECH HC-06 que nos permite conectar a ChipKIT mediante bluetooth con otros dispositivos. Para utilizarlo disponemos de la librería “SoftwareSerial.h” con la que podremos utilizar cualquier pin de nuestra placa para realizar comunicación serie necesaria para este módulo.

Figura 61: Módulo DSD TECH HC-06

2.21 Bloque de aviso a servicios de emergencia

Por último, y uno de los bloques más importante, es el que nos va a permitir avisar a los servicios de emergencia, número 112 en Europa. Para esto, utilizaremos la tarjeta GSM/GPRS basada en el módulo SIM900. Esta tarjeta, compatible con Arduino, nos permite enviar y recibir llamadas y SMS e incluso conectarnos a internet.

Figura 62: Tarjeta GSM/GPRS basada en el módulo SIM900

Gracias a este módulo seremos capaces de avisar a los servicios de emergencias y de enviarles el informe pertinente sobre lo ocurrido para ellos tomen la última decisión.

2.22 Posibles mejoras

A pesar de que este prototipo funcionaria tal y como se especifica a lo largo de todo el documento, existen posibles mejoras que podríamos realizarle para que sea un dispositivo aún más seguro y fiable. Además, pueden surgir dudas de que pasaría si alguno de los elementos fallara o si la toma de decisiones falla etc.

A continuación, se van a exponer alguna posible mejora al sistema estudiado.

El primer lugar donde podríamos mejorar es en el interior de nuestro sistema, es decir, ChipKIT y todos los módulos que le conectamos. Como se comentó, ChipKIT es una placa de desarrollo muy fácil de utilizar y conseguir, pero es para uso no profesional, aunque se puede utilizar para aplicaciones pequeñas. Además, al tener tantas piezas diferentes, es más posible que alguna

de ellas falle con lo que nuestro sistema ya no realizaría sus funciones correctamente. Aunque tomáramos muchas precauciones en su realización, sería mucho más fácil que fallara que, por ejemplo, un SoC que desarrollemos nosotros mismos especialmente diseñado para esta función.

Figura 63: SoC de Advanced Micro Devices

El SoC que desarrollemos puede incluir todas las funcionalidades que requerimos de los módulos que hemos instalado y, además, podemos diseñarlo incluyendo más características que, utilizando los módulos ya creados por otra persona, no tendríamos. Lo único que no podríamos añadir a este SoC serían las antenas para las comunicaciones o el altavoz, pero, aun añadiendo esos extras a la placa donde irá este SoC, se reducen los componentes enormemente respecto al diseño planteado.

El siguiente problema que se plantea es ¿qué pasaría si alguno de los módulos o sensores fallara? En este estudio no se ha considerado esa posibilidad por lo que será un punto donde mejorar.

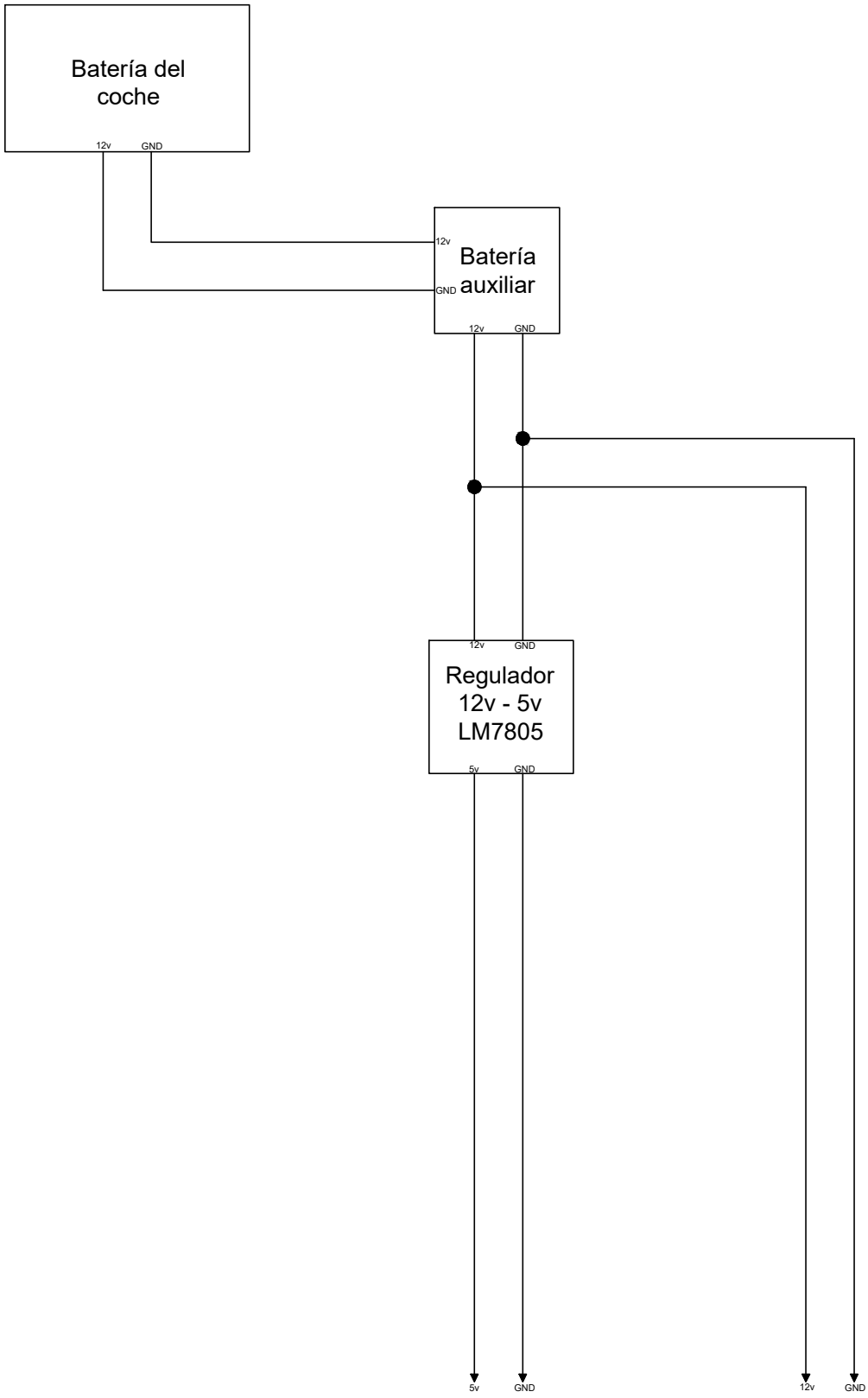
Una posible solución a esto sería que el sistema se haga auto chequeos periódicos para cerciorarse de que todo está funcionando correctamente y, en caso de que algún modulo o sensor no esté en perfecto estado, el sistema deberá avisar al conductor de que el sistema de seguridad no está en perfectas condiciones, por ejemplo, con un piloto que se le encienda en el salpicadero. El coche seguiría funcionando correctamente pero el conductor ya sabrá que el sistema de seguridad no está bien y debe llevarlo a reparar.

Otro problema que se puede plantear es ¿qué pasaría si la red neuronal artificial que toma las decisiones falla en su decisión y llama a los servicios de emergencia sin ser necesario?

Una solución a este problema podría ser la de enviar a los servicios de emergencia en que porcentaje de seguro esta la RNA de que lo ocurrido ha sido un accidente. De esta manera, la última decisión la tomarían los servicios de emergencia por lo que aun sería más acertada la actuación necesaria para esa situación. Incluso, se podría crear un sistema de comunicación dentro del SoC para que los propios servicios de emergencia se comunicaran con el vehículo que se ha puesto en contacto con ellos para valorar la situación.

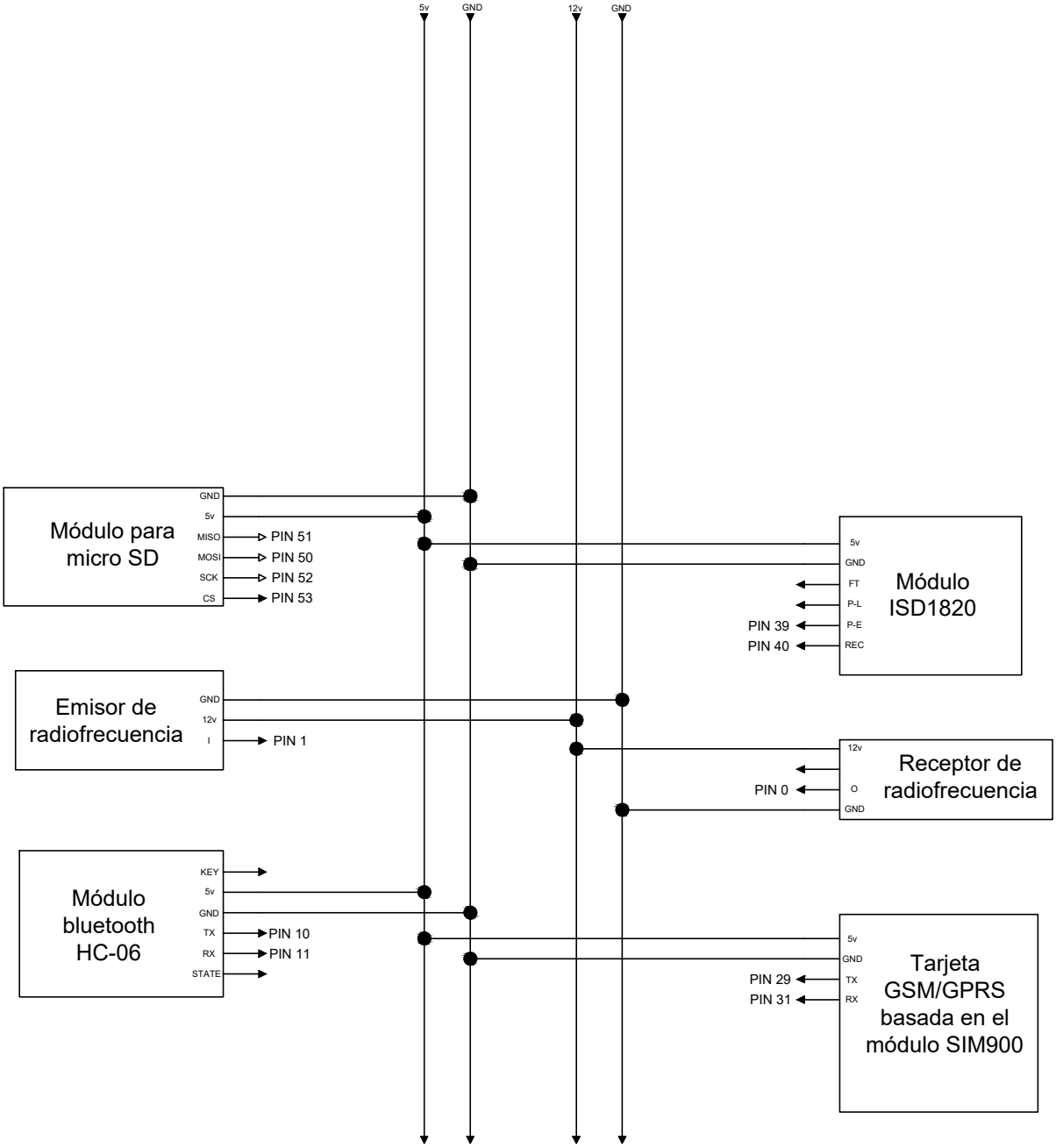
Para tener un informe todavía más completo cuando se produzca el accidente, se podrían leer, a parte de los sensores del coche que ya se mencionan en el documento, otros que, aunque igual no fueran relevantes para detectar un accidente, sí que pueden ser relevantes para que se interprete mejor que es lo que realmente ocurrió cuando se produjo el accidente.

3 Planos



	Fecha	Nombre		ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL Universidad de la Rioja GRADO DE INGENIERÍA EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA	
Dibujado	03/09/19	Eder Villar			
Comprob.	04/09/19	Javier Vicuña			
D.S.Normas	U.N.E	Tolerancia General			

Escala: S/E	TFG	Codigo de referencia: Numero de plano: 1
Proyección	ESQUEMA DE CONEXION DE LA ALIMENTACION AUXILIAR CON SALIDAS A 12V Y 5V	Sustituye a: Eder Villar Sustituido por: Eder Villar



	Fecha	Nombre		ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INDUSTRIAL Universidad de la Rioja GRADO DE INGENIERÍA EN ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA
Dibujado	03/09/19	Eder Villar		
Comprob.	04/09/19	Javier Vicuña		
D.S.Normas	U.N.E	Tolerancia General		

Escala: S/E	TFG	Codigo de referencia: Numero de plano: 2
Proyección	**ESQUEMA DE CONEXION DE LOS MODULOS A CHIPKIT MAX 32**	Sustituye a: Eder Villar Sustituido por: Eder Villar

4 Pliego de condiciones

4.1 Vehículo

El vehículo en el que se instale este dispositivo debe poseer las siguientes características:

- Las comunicaciones entre los sensores del vehículo se deben realizar por bus CAN y el sistema que se va a instalar debe ser capaz de entender el protocolo que utiliza ese modelo de coche.
- El coche debe contar con un máximo de dos buses CAN o, en su defecto, que los sensores que vayamos a utilizar estén conectados a un máximo de dos buses
- Debe montar, por lo menos, un airbag.
- Debe contener como mínimo los sensores que se indican en el apartado 2.15.7.1 de este documento. En caso de que falte alguno se deberían instalar manualmente utilizando sensores de la propia desarrolladora del vehículo.
- Los sensores del apartado 2.15.7.2 será necesario testearlos para comprobar si es necesario instalarlos o no. Además, en caso de que se instalen deben ser correctamente protegidos para soportar las condiciones que se vayan a dar en la zona donde se coloquen.
- Debe contener al menos una batería de 12v.

4.2 Placa de desarrollo

La placa de desarrollo será un ChipKIT max32 de la marca Digilent. Se puede utilizar esta placa o una de gama superior siempre que cumpla con las especificaciones propuestas en el apartado 2.16 de este documento.

4.3 Adquisición de datos

La adquisición de datos la llevará a cabo la propia placa de desarrollo y debe leer todos los sensores del coche por medio de los buses CAN presentes en este. Todos los sensores se leen por el bus CAN excepto el sensor de ritmo cardiaco que será una pulsera que porte consigo el conductor que se comunicará por bluetooth con nuestra placa.

Todos estos datos que se reúnen deben ser guardados en la memoria volátil del microprocesador hasta que se requiera otra acción.

4.4 Red neuronal

Lo primero que se debe hacer es preparar la información con la que se va a alimentar la red ya que enviar a la red valores instantáneos no es útil porque lo que realmente nos indica que algo raro está ocurriendo es que se haya producido un cambio brusco en los sensores.

Para ello, nuestro microprocesador deberá realizar un cálculo antes de presentarle los datos a la RNA. El procedimiento será el siguiente

- Realizará la media de las ultimas muestras tomadas de los sensores
- Guardará esta media
- Realizará la media de las siguientes muestras que se tomen
- Restará ambas medias y hará el valor absoluto del resultado

Realizando este tratamiento de los datos, nos aseguramos de que tendremos variaciones claras en los datos para que la red sea capaz de tomar decisiones en base a cambios que se producen en los sensores.

El número de muestras de las cuales se hará la media dependerá del tiempo de muestreo que se configure. Si este es pequeño, la media se deberá realizar de más muestras y si es grande, la media se realizará de menos muestras.

Para el diseño, desarrollo y entrenamiento de la red neuronal se utilizará cualquier software y/o librería que nos permita esto. No se deberá utilizar el propio microprocesador para entrenar la red ya que para estas tareas es limitado.

Para este diseño se recomienda utilizar la toolbox de Deep Learning para Matlab. En caso de utilizarla, los requisitos mínimos del ordenador en el que se instale son los siguientes:

Sistema operativo

  - **Windows:** Windows 7 SP1 en adelante, Windows Server 2008 SP2 en adelante
  - **MAC:** macOS 10.10 – 10.11
  - **Linux:** Kernel 2.6 o superior, glibc 2.11 o superior
- **Procesador:** Intel o AMN x86-64 con soporte para instrucciones AVX2
- **Disco:** 4 – 6GB de almacenamiento libres
- **RAM:** 1GB mínimo, 4GB recomendado
- **Tarjeta gráfica:** Soporte para OpenGL 3.3 recomendado con 1GB en GPU

La red tendrá una capa de entrada con un mínimo de ocho neuronas, una por cada uno de los sensores que se ha comentado en el apartado 2.15.7.1. Las

neuronas necesarias para los sensores del apartado 2.15.7.2 dependerán del número de estos que vayamos a instalar. La capa de salida contara con 4 neuronas, una por cada nivel de clasificación. Además, las neuronas de esta última capa tendrán como función de activación la función sigmoidea que nos dará resultados entre 0 y 1.

Estos 4 niveles de clasificación son:

- Funcionamiento normal: El coche está en perfectas condiciones.
- Accidente leve: Se ha producido una colisión a baja velocidad.
- Accidente moderado: El coche a sufrido una colisión en la que los sensores han detectado una variación moderada con respecto al estado o los estados anteriores.
- Accidente grave: Los sensores han detectado una gran variación respecto al estado o estados anteriores.

La metodología para el diseño de las capas internas de la red se ha expuesto en la parte final del apartado 2.16.3 de este documento. Esta es una recomendación basada en metodologías de expertos en la materia, pero el diseñador puede utilizar sus propios métodos.

4.5 Alimentación de ChipKIT y módulos

La alimentación de la placa que incluye el microprocesador y la de los módulos cercanos se deberá realizar con la batería del propio coche. Se deberá pasar esta por un regulador para bajar la tensión de 12v a 5v que es con la que trabajamos.

Exceptuando los módulos de transmisión y recepción de frecuencia que irán alimentados a 12v el resto irá a 5v.

Se deberá instalas una batería auxiliar de 2000mAh como mínimo con salida de 12v para evitar cortes de alimentación en caso de fallo en la batería del coche. Estos 12v pasaran por un transformador que baje esta tensión hasta los 5v para alimentar la mayoría de los módulos. Los que van a 12v, los de comunicación por radiofrecuencia, irán directamente conectados a esta batería.

4.6 Compartimento de seguridad

La placa ChipKIT junto a todos sus módulos y la batería auxiliar deberán ser aisladas en una caja para protegerlos. Esta caja, que la llamaremos caja negra al igual que la de los aviones, debe ser muy resistente a los impactos y debe ser completamente ignifuga. Deberá ser construida con acero de alta densidad y deberá estar pintada con un color muy visible el cual también deberá ser resistente a las llamas.

Así mismo, como alguno de los módulos que completan este sistema llevan antenas para la comunicación externa, este compartimento debe ser capaz de no bloquear estas comunicaciones, si no, se deberán proteger en un compartimento aparte que si permita esa comunicación.

4.7 Conexión de ChipKIT y módulos

Se deberá diseñar un PCB en el que irán todos los módulos unidos a ChipKIT para evitar desconexión de cables y crear un sistema más robusto. La conexión a bus CAN se dejará como unos puntos de conexión al PCB donde luego se conectará al bus CAN del coche en el que se instale.

4.8 Módulos

Los módulos necesarios para conectar a nuestro ChipKIT max32 se especifican en los apartados 2.18, 2.19, 2.20. 2.21.

En el 2.19, que se especifica que módulo para comunicación por radiofrecuencia utilizar, también se habla sobre la antena que deberíamos instalar para obtener un alcance mayor.

4.9 Condiciones térmicas

Debido a que el sistema estará dentro de un compartimento sellado, debe instalarse algún sistema de refrigeración de este puesto que se podrían quemar los componentes internos.

4.10 Normativa aplicable

La normativa y leyes aplicables para llevar a cabo este prototipo se especifican en los apartados 2.8 y 2.9 de este documento.

4.11 Certificación CE

Debido a que este producto podría ser comercializado en Europa necesita de esta certificación. Por ello, es obligatorio que se mande a un centro de certificación para que del sello de CE el cual deberá ser puesto en el sistema cumpliendo las siguientes reglas:

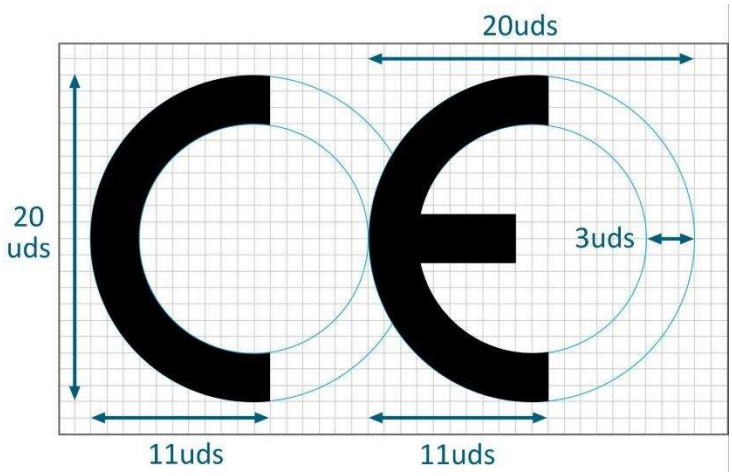


Figura 64: Logo de la certificación CE con sus medidas

- Deben conservarse las proporciones, siendo la dimensión vertical mínima de 5 mm.
- Debe colocarse sobre el producto o su placa descriptiva. Cuando no sea posible, deberá fijarse al embalaje si lo hubiera y en los documentos que lo acompañan, si la Directiva lo exige.
- Se colocará de forma visible, legible e indeleble.
- Debe ir seguida del número/s de identificación del Organismo/s Notificado/s involucrado/s en su caso.
- Es el único marcado que indica que el producto cumple las Directivas de aplicación.
- Debe colocarse al final de la fase de control de producción.
- Lo fijará el fabricante o su representante autorizado dentro de la Unión Europea. Excepcionalmente, cuando la Directiva lo permita, podrá fijarlo la persona responsable de la puesta en el mercado del producto en la Unión Europea.
- Está prohibido colocar signos que puedan confundirse con el marcado "CE", tanto en significado como en la forma. Un producto podrá llevar otras marcas o sellos, siempre que no se confundan con el marcado "CE" y que no reduzcan la legibilidad y visibilidad de éste. Los fabricantes que tengan marcas susceptibles de confundirse con el marcado "CE", están autorizados a poseer su marca durante 10 años después de la adopción del reglamento si estas marcas han sido registradas antes del 30/06/89 y están actualmente en servicio.

5 Presupuesto

1.1. Unidades de proyecto

Hardware	
ChipKIT y módulos	
Nombre	Descripción
ChipKIT max32	Placa base con procesador de 32 bits
MOD-MSDC	Módulo para lectura y escritura de microSD
ISD1820 con altavoz	Módulo grabador y reproductor de audios
Boton	Pulsador mecanico
FS1000A	Módulo emisor de radiofrecuencia
XY-MK-5v	Módulo receptor de radiofrecuencia
Tarjeta GSM/GPRS	Tarjeta para envio y recepcion de mensajes y llamadas
Alimentacion auxiliar	
Nombre	Descripción
Bateria Li-Po	Bateria de litio-polímero de 2000mAh
LM8705	Regulador de continua 12v a 5v
Sensores	
Nombre	Descripción
Giñada	Medidor de la velocidad angular en el eje vertical
Inclinación	Medidor de la inclinación en un eje
Acelerómetro	Medidor del cambio de velocidad en una dirección
Colisión	Medidor de impacto
GPS	Dispositivo de geolocalización
Presion en asiento	Medidor de presion en el asiento del automovil
Medidor de puslo	Pulsera medidora del ritmo cardiaco
KY-026	Sensor de llama
Galga RS-PRO	Sensor de deformación
Sensor de giro	Sensor de angulo de giro para la dirección

Software	
Nombre	Descripción
MatLab	Software de calculo
Deep Learning Toolbox	Modulo para creacion y entrenamiento de redes neuronales
Arduino IDE	Software para programar ChipKIT

Mano de obra	
Nombre	Descripción
Investigación	Tiempo dedicado a la investigación
Documentación	Tiempo dedicado a la búsqueda de documentación
Redacción	Tiempo dedicado a la redaccion del documuento

1.2. Precios unitarios

Hardware		
ChipKIT y módulos		
Nombre	Descripción	Precio [€]
ChipKIT max32	Placa base con procesador de 32 bits	60,65
MOD-MSDC	Módulo para lectura y escritura de microSD	11,99
ISD1820 con altavoz	Módulo grabador y reproductor de audios	3,31
Boton	Pulsador mecanico	0,24
FS1000A	Módulo emisor de radiofrecuencia	0,41
XY-MK-5v	Módulo receptor de radiofrecuencia	0,41
Tarjeta GSM/GPRS	Tarjeta para envio y recepcion de mensajes y llamadas	22,99
Alimentacion auxiliar		
Nombre	Descripción	Precio [€]
Bateria Li-Po	Bateria de litio-polímero de 2000mAh	11,5
LM7805	Regulador de continua 12v a 5v	1,32
Sensores		
Nombre	Descripción	Precio [€]
Giñada	Medidor de la velocidad angular en el eje vertical	101,98
Inclinación	Medidor de la inclinación en un eje	29,94
Acelerómetro	Medidor del cambio de velocidad en una dirección	27,91
Colisión	Medidor de impacto	15,89
GPS	Dispositivo de geolocalización	42,28
Presion en asiento	Medidor de presion en el asiento del automovil	10,66
Medidor de puslo	Pulsera medidora del ritmo cardiaco	18,59
KY-026	Sensor de llama	2,49
Galga RS-PRO	Sensor de deformación	9,34
Sensor de giro	Sensor de angulo de giro para la dirección	126,97

Software		
Nombre	Descripción	Precio [€]
MatLab	Software de calculo	2000
Deep Learning Toolbox	Modulo para creacion y entrenamiento de redes neuronales	1150
Arduino IDE	Software para programar ChipKIT	0

Mano de obra		
Nombre	Descripción	Precio [€/h]
Investigación	Tiempo dedicado a la investigación	25
Documentación	Tiempo dedicado a la búsqueda de documentación	25
Redacción	Tiempo dedicado a la redaccion del documuento	25

1.3. Mediciones

Hardware		
ChipKIT y módulos		
Nombre	Descripción	Unidades
ChipKIT max32	Placa base con procesador de 32 bits	1
MOD-MSDC	Módulo para lectura y escritura de microSD	1
ISD1820 con altavoz	Módulo grabador y reproductor de audios	1
Boton	Pulsador mecanico	2
FS1000A	Módulo emisor de radiofrecuencia	1
XY-MK-5v	Módulo receptor de radiofrecuencia	1
Tarjeta GSM/GPRS	Tarjeta para envio y recepcion de mensajes y llamadas	1
Alimentacion auxiliar		
Nombre	Descripción	Unidades
Bateria Li-Po	Bateria de litio-polímero de 2000mAh	1
LM7805	Regulador de continua 12v a 5v	1
Sensores		
Nombre	Descripción	Unidades
Giñada	Medidor de la velocidad angular en el eje vertical	1
Inclinación	Medidor de la inclinación en un eje	2
Acelerómetro	Medidor del cambio de velocidad en una dirección	2
Colisión	Medidor de impacto	1
GPS	Dispositivo de geolocalización	1
Presion en asiento	Medidor de presion en el asiento del automovil	2
Medidor de puslo	Pulsera medidora del ritmo cardiaco	1
KY-026	Sensor de llama	1
Galga RS-PRO	Sensor de deformación	1
Sensor de giro	Sensor de angulo de giro para la dirección	1

Software		
Nombre	Descripción	Unidades
MatLab	Software de calculo	1
Deep Learning Toolbox	Modulo para creacion y entrenamiento de redes neuronales	1
Arduino IDE	Software para programar ChipKIT	1

Mano de obra		
Nombre	Descripción	Horas
Investigación	Tiempo dedicado a la investigación	150
Documentación	Tiempo dedicado a la búsqueda de documentación	50
Redacción	Tiempo dedicado a la redaccion del documuento	100

1.4. Presupuestos parciales

Hardware			
ChipKIT y módulos			
Nombre	Descripción	Precio [€]	Subtotal [€]
ChipKIT max32	Placa base con procesador de 32 bits	60,65	60,65
MOD-MSDC	Módulo para lectura y escritura de microSD	11,99	72,64
ISD1820 con altavoz	Módulo grabador y reproductor de audios	3,31	75,95
Boton	Pulsador mecanico	0,24	76,19
FS1000A	Módulo emisor de radiofrecuencia	0,41	76,6
XY-MK-5v	Módulo receptor de radiofrecuencia	0,41	77,01
Tarjeta GSM/GPRS	Tarjeta para envio y recepcion de mensajes y llamadas	22,99	100
Alimentacion auxiliar			
Nombre	Descripción	Precio [€]	Subtotal [€]
Bateria Li-Po	Bateria de litio-polímero de 2000mAh	11,5	11,5
LM7805	Regulador de continua 12v a 5v	1,32	12,82
Sensores			
Nombre	Descripción	Precio [€]	Subtotal [€]
Giñada	Medidor de la velocidad angular en el eje vertical	101,98	101,98
Inclinación	Medidor de la inclinación en un eje	29,94	131,92
Acelerómetro	Medidor del cambio de velocidad en una dirección	27,91	159,83
Colisión	Medidor de impacto	15,89	175,72
GPS	Dispositivo de geolocalización	42,28	218
Presion en asiento	Medidor de presion en el asiento del automovil	10,66	228,66
Medidor de puslo	Pulsera medidora del ritmo cardiaco	18,59	247,25
KY-026	Sensor de llama	2,49	249,74
Galga RS-PRO	Sensor de deformación	9,34	259,08
Sensor de giro	Sensor de angulo de giro para la dirección	126,97	386,05
Total		498,87 €	

Asciende el citado apartado de Hardware a la cantidad de CUATROCIENTO NOVENTA Y OCHO EUROS CON OCHENTA Y SIETE CÉNTIMOS.

Software			
Nombre	Descripción	Precio [€]	Subtotal [€]
MatLab	Software de calculo	2000	2000
Deep Learning Toolbox	Modulo para creacion y entrenamiento de redes neuronales	1150	3150
Arduino IDE	Software para programar ChipKIT	0	3150
Total		3.150 €	

Asciende el citado apartado de Software a la cantidad de TRES MIL CIENTO CINCUENTA EUROS.

Mano de obra			
Nombre	Descripción	Precio [€/h]	Subtotal [€]
Investigación	Tiempo dedicado a la investigación	25	3750
Documentación	Tiempo dedicado a la búsqueda de documentación	25	5000
Redacción	Tiempo dedicado a la redaccion del documuento	25	7500
Total		7.500 €	

Asciende el citado apartado de Mano de obra a la cantidad de SIETE MIL QUINIENTOS EUROS.

1.5. Presupuesto total

Concepto	Precio [€]	Subtotal [€]
Hardware	498,87 €	498,87 €
Software	3.150 €	3.648,87 €
Mano de obra	7.500 €	11.148,87 €
Total	11.148,87 €	22.297,74 €
Total con IVA (21%)	2.341,26 €	24.639,00 €
Total con beneficio industrial (6%)	668,93 €	25.307,93 €
Total	25.307,93 €	

Asciende el trabajo estudiado a la cantidad total de VEINTICINCO MIL TRESCIENTOS SIETE EUROS CON NOVENTA Y TRES CÉNTIMOS.

Notas:

- → Se han utilizado precios de agosto de 2019. En el supuesto que se desee utilizar este TFG transcurridos 18 meses se recomienda la actualización de precios.
- → En el presupuesto se ha incluido una licencia de Matlab y una de la toolbox para Deep Learning. En el caso de que se vaya a usar un software de código libre se restara al presupuesto el precio de ese apartado.
- → Se ha incluido el software recomendado para la programación de ChipKIT. En caso de utilizar otro, se deberá actualizar el apartado en el que se incluye.
- → Los sensores que están sujetos a testeo solo se han añadido al presupuesto como unidades únicas. Si se van a instalar más de uno, deberán añadirse al correspondiente apartado